Universidad de Valladolid
Facultad de Ciencias

Trabajo Fin de Grado

Grado en Matemáticas

Registros lineales retroalimentados y su aplicación a la criptografía en flujo

Autor: Pablo Hervás García

Tutor/es: José Enrique Marcos Naveira

Índice general

1. Preliminares matemáticos **3**
1.1. Orden de un polinomio . . . 3
1.2. Caracteres de un grupo . . . 7

2. Introducción a la Criptografía **11**
2.1. Criptosistemas y ataques . . . 11
2.2. Criptosistemas simétricos y asimétricos . . . 15
2.3. Libretas de un solo uso . . . 16
2.4. Distinción entre cifrado en bloque y cifrado en flujo . . . 17
2.5. Fundamentos del cifrado en bloque . . . 18
2.6. Fundamentos del cifrado en flujo . . . 19
2.7. AES, ¿para qué son necesarios los cifrados en flujo? . . . 21

3. Sucesiones pseudoaleatorias **23**
3.1. Bloques y rachas . . . 24
3.2. Desequilibro respecto a un carácter . . . 26
3.3. Autocorrelación . . . 27
3.4. Sucesiones de de Bruijn . . . 28
3.5. Propiedad de desplazamiento y suma . . . 31

4. Registros de Desplazamiento con Retroalimentación Lineal (LFSRs) **33**
4.1. Introducción . . . 33
4.2. LFSRs y complejidad lineal . . . 35
4.3. Forma matricial . . . 39
4.4. Periodo de un LFSR . . . 41
4.5. Propiedades de las m-sucesiones . . . 44
4.6. Algortimo de Berlekamp-Massey . . . 45

5. Aplicaciones de los LFSRs en Criptografía **50**
5.1. Combinadores no lineales . . . 50
5.2. Filtrado no lineal . . . 52
5.3. Otros esquemas . . . 53
5.4. Cifrados en flujo basados en LFSR con uso actual . . . 55

A. Teoría de cuerpos **59**
A.1. Extensión de cuerpos . . . 59
A.2. Cuerpos finitos . . . 64

Bibliografía **70**

Capítulo 1

Preliminares matemáticos

> There is no branch of mathematics, however abstract, which may not some day be applied to phenomena of the real world.
>
> *Nikolai Lobachevsky*

En este primer Capítulo introduciremos algunos conceptos que, si bien sencillos, no quedan recogidos en las asignaturas obligatorias de la carrera.
El motivo para introducirlos con antelación es poder proporcionar una profundidad y contextualización adecuada, sin que por otro lado sea necesario detenernos en exceso cuando aparezcan posteriormente.
La primera sección viene recogida en [25][Capítulo 3], mientras que la segunda proviene de [16][Apéndice A].

1.1. Orden de un polinomio

Si queremos definir "el orden de un polinomio", sería natural que estuviera relacionado con el orden (multiplicativo) de sus raíces. La Definición 1.2 es compatible con esta idea, pero solo se entenderá bien la motivación precisa detrás de ella cuando lleguemos al Capítulo de Linear Feedback Shift Registers.

Proposición 1.1. *Dado $g \in \mathbb{F}_q[x]$ un polinomio de grado n con $x \nmid g$, existe un e natural tal que $g \mid (x^e - 1)$. Además, se puede elegir $e \leq q^n - 1$.*

> ***Demostración:***
> Decir que $g \mid (x^e - 1)$ significa lo mismo que decir $x^e \equiv 1 \pmod{g}$. Como $\gcd(x, g) = 1$, $\bar{x}$ es un elemento invertible de $\mathbb{F}_q[x]/(g)$. Por tanto, $\bar{x}$ genera un grupo multiplicativo $\langle \bar{x} \rangle$, y el menor e será entonces el orden de dicho grupo. Notemos que en este caso $e \leq q^n - 1$, pues solo hay $q^n - 1$ elementos no nulos en $\mathbb{F}_q[x]/(g)$. □

Por otro lado, si $x \mid g$, obviamente ningún e funciona. Pero es posible extender la definición a esos casos sin mayor inconveniente.

Definición 1.2. Sea $f \in \mathbb{F}_q[x]$ un polinomio no nulo, siempre se puede escribir $f = x^m g$ de forma que $x \nmid g$. Entonces se define el orden de f como el menor e natural tal que $g \mid (x^e - 1)$, o equivalentemente, como el orden de $\bar{x}$ en el anillo $\mathbb{F}_q[x]/(g)$.

Nota 1.3. Si $q = p^m$ (esta "m" no tiene nada que ver con la de x^m), está claro que el orden de un polinomio $f \in \mathbb{F}_q[x]$ no cambia si se le considera en otro anillo $\mathbb{F}_{q'}[x]$ con $q' = p^{m'}$ (por

supuesto, los coeficientes de f deben estar en $\mathbb{F}_{q'}$ para poder hacer eso). El orden de f solo depende de la característica p del cuerpo finito subyacente $\mathbb{F}_p$. Esta dependencia se ha omitido en la definición, pues entorpece innecesariamente la notación. Nosotros siempre supondremos que $\mathbb{F}_p$ permanece fijo.

Ahora procederemos a estudiar cómo se puede determinar el orden de un polinomio. Primero necesitamos un resultado auxiliar.

Lema 1.4. *Sea $f \in \mathbb{F}_q[x]$ un polinomio no nulo, para todo t natural se tiene que $f \mid (x^t - 1)$ si y solo si* $\mathrm{ord}(f) \mid t$.

Demostración:

$\Rightarrow$ Sea $e = \mathrm{ord}(f)$. Escribimos $t = qe + r$ con $0 \leq r < e$, y queda $x^t - 1 = (x^{qe} - 1)x^r + (x^r - 1)$. Por definición, $f \mid (x^e - 1)$, y como $(x^e - 1) \mid (x^{qe} - 1)$, obtenemos que $f \mid (x^{qe} - 1)$. Esto, junto a que $f \mid (x^t - 1)$ por hipótesis, implica que $f \mid (x^r - 1)$. Como $r < e = \mathrm{ord}(f)$, debe ser $r = 0$.

$\Leftarrow$ Recíprocamente, si $e \mid t$, tenemos que $(x^e - 1) \mid (x^t - 1)$, y como $f \mid (x^e - 1)$, obtenemos que $f \mid (x^t - 1)$.

□

Los tres resultados siguientes, la Proposición 1.5, la Proposición 1.8, y la Proposición 1.11, proporcionan, de menor a mayor generalidad, el orden explícito de un polinomio. Veremos que solo es necesario conocer la factorización de dicho polinomio en factores irreducibles.

Proposición 1.5. *Sea $f \in \mathbb{F}_q[x]$ un polinomio (no nulo) irreducible de grado n. Entonces* $\mathrm{ord}(f)$ *es igual al orden multiplicativo de cualquiera de las raíces de f en $\mathbb{F}_{q^n}$.*

Demostración:
Sabemos que f factoriza completamente sobre $\mathbb{F}_{q^n}$, sea α una raíz de f en $\mathbb{F}_{q^n}$. Ya se probó en la Proposición A.9 que existe un isomorfismo entre $\mathbb{F}_q[\alpha]$ y $\mathbb{F}_q[x]/(f)$, y en la Demostración de la proposición citada se construía dicho isomorfismo de forma que mandaba α a $\bar{x}$. Con este isomorfismo, $\alpha^e = 1$ si y solo si $\bar{x}^e = \bar{1}$, lo que equivale a decir que $f \mid (x^e - 1)$. En consecuencia, el orden de α y el orden de f deben coincidir (en particular, todas las raíces de f tienen el mismo orden, un hecho que no comentamos antes, pero se podría haber deducido de la Proposición A.27). □

Ejemplo 1.6. Consideramos el polinomio $f = x^4 + x + 1$ en $\mathbb{F}_2[x]$. Es fácil comprobar que f es irreducible, ya que f no tiene raíces en $\mathbb{F}_2$ (solo hay que comprobar $f(0) \neq 0$ y $f(1) \neq 0$) y $f \neq (x^2 + x + 1)^2$ ($x^2 + x + 1$ es el único polinomio irreducible de grado 2 en $\mathbb{F}_2[x]$). El orden de f debe dividir a $2^4 - 1 = 15$, pero $f \nmid (x^5 - 1)$, así que concluimos que $\mathrm{ord}(f) = 15$.

Ejemplo 1.7. Consideramos ahora el polinomio $f = x^4 + x^3 + x^2 + x + 1$ en $\mathbb{F}_2[x]$. Al igual que en el Ejemplo anterior, es fácil comprobar que f es irreducible. Pero en este caso $(x-1)f = x^5 - 1$, en consecuencia $\mathrm{ord}(f) = 5$.

Proposición 1.8. *Sea $f \in \mathbb{F}_q[x]$ un polinomio (no nulo) que es potencia $f = g^b$ con $b \geq 2$ de otro polinomio $g \in \mathbb{F}_q[x]$, con g irreducible y $x \neq g$. Sea t el menor natural con $p^t \geq b$, donde p es la característica de $\mathbb{F}_q$. Entonces* $\mathrm{ord}(f) = \mathrm{ord}(g)p^t$.

Demostración:
Sea $e = \text{ord}(f)$, $c = \text{ord}(g)$. Vamos a probar primero que todas las raíces de $x^c - 1$ son simples. Sea n el grado de g, sabemos que g factoriza completamente sobre $\mathbb{F}_{q^n}$. En consecuencia, todas las raíces de g, que como $x \neq g$ no son 0, son también raíces de $x^{q^n-1}-1$. Por la Proposición 1.5, c es el orden de estas raíces, así que $c \mid q^n - 1$, en particular $p \nmid c$. Con esto, si consideramos $h = x^c - 1$, tenemos $h' = cx^{c-1} \neq 0$, por tanto $\gcd(h, h') = 1$ y todas las raíces de h son simples.
Ahora daremos una factorización parcial de e. Como $f \mid (x^e - 1)$, también $g \mid (x^e - 1)$, así que $c \mid e$. Por otra parte, como $f = g^b$, tenemos $f \mid (x^c - 1)^b$, y por hipótesis, $p^t \geq b$, lo que implica $f \mid (x^c - 1)^{p^t} = x^{cp^t} - 1$. Así que $e \mid cp^t$, también $c \mid e$, y recordemos p es primo y que $p \nmid c$: de estas cuatro relaciones se deduce que e es de la forma $e = cp^u$ con $0 \leq u \leq t$.
Falta ver que $t = u$. Notemos que $f = g^b$ y $c = \text{ord}(g)$, y que todas las raíces de $x^c - 1$ son simples, luego las raíces de f serán raíces c-ésimas de la unidad con multiplicidad b. Con la expresión anterior para e, tenemos $f \mid x^e = x^{cp^u} - 1 = (x^c - 1)^{p^u}$, por lo que cualquier raíz de f será una raíz c-ésima de la unidad con multiplicidad a lo sumo p^u. Por esta razón, debe ser $p^u \geq b$, por lo que $u \geq t$ y hemos acabado. □

Observación 1.9. Si el polinomio $f \in \mathbb{F}_q[x]$ es una potencia de x, su orden es 1 por definición.

Ejemplo 1.10. Consideramos el polinomio $f = x^6 + x^5 + x^3 + x + 1$ en $\mathbb{F}_2[x]$. Tenemos que $f = (x^2 + x + 1)^3$, donde $g = x^2 + x + 1$ es un polinomio irreducible. Ya sabemos que $\text{ord}(g) = 2^2 - 1 = 3$, y $b = 2$ es el menor natural tal que $p^b \geq 3$, así que el orden de f es $\text{ord}(f) = 2^2 \cdot 3 = 12$.

Proposición 1.11. *Sea $f \in \mathbb{F}_q[x]$ un polinomio no nulo, siempre se puede escribir $f = g_1 \dots g_k$ de forma que los $g_1, \dots, g_k$ sean coprimos. Entonces* $\text{ord}(f) = \text{lcm}(\text{ord}(g_1), \dots, \text{ord}(g_k))$.

Demostración:
Sin pérdida de generalidad $x \nmid g_i$ para todo i. Sea $e = \text{ord}(f)$, $c_i = \text{ord}(g_i)$, $c = \text{lcm}(c_1, \dots, c_k)$. Cada $c_i \mid c$, así que $g_i \mid x^c - 1$. Como los g_i son coprimos, entonces también $f = g_1 \dots g_k \mid x^c - 1$ y por tanto $e \mid c$. Por otro lado, como $f = g_1 \dots g_k$, cada $g_i \mid x^e - 1$, por lo que cada $c_i \mid e$ y en consecuencia $c \mid e$. □

Ejemplo 1.12. Consideramos $f = x^4 + x^3 + x + 1$ en $\mathbb{F}_2[x]$. Es sencillo factorizar f, obtenemos $f = (x + 1)^2(x^2 + x + 1)$, donde $\text{ord}((x + 1)^2) = 2^1 \cdot 1 = 2$ y $\text{ord}(x^2 + x + 1) = 2^2 - 1 = 3$. Entonces $\text{ord}(f) = \text{lcm}(2, 3) = 6$.

Finalmente llegamos a la Definición y el Teorema que desempeñarán un papel esencial en el siguiente Capítulo.

Definición 1.13. Un polinomio $f \in \mathbb{F}_q[x]$ de grado $n \geq 1$ se dice que es primitivo sobre $\mathbb{F}_q$ si es el polinomio mínimo sobre $\mathbb{F}_q$ de un elemento primitivo de $\mathbb{F}_{q^n}$. Un polinomio $h \in \mathbb{F}_q[x]$ se dice que es primitivo salvo constante sobre $\mathbb{F}_q$ si existe $a \in \mathbb{F}_q$ tal que ah es primitivo.

Observación 1.14. A la vista de la Proposición 1.1 y la Proposición 1.5, los polinomios primitivos salvo constante tienen orden máximo entre todos los polinomios irreducibles con grado igual o menor. También a la vista de estos resultados, se tiene que un polinomio $f \in \mathbb{F}_q[x]$ es primitivo si y solo si $\bar{x}$ genera el grupo multiplicativo de $\mathbb{F}_q[x]/(f)$.

El Teorema 1.16 es aún más fuerte. Primero tenemos que tratar aparte un molesto y trivial caso particular.

Observación 1.15. Si $q = 2$ y $n = 1$, un polinomio $f \in \mathbb{F}_q[x]$ de grado n es primitivo salvo constante sobre $\mathbb{F}_q$ si y solo si $f = x - 1$. El polinomio $f = x$ tiene el mismo grado y orden que el anterior, pero no es primitivo. El polinomio $f = 1$ tiene menor grado y el mismo orden que los dos primeros, pero tampoco es primitivo.

Teorema 1.16. *Si $q > 2$ o $n > 1$, un polinomio $f \in \mathbb{F}_q[x]$ de grado n es primitivo salvo constante sobre $\mathbb{F}_q$ si y solo si* $\mathrm{ord}(f) = q^n - 1$. *En particular, los polinomios primitivos salvo constante tienen orden máximo entre todos los polinomios con grado igual o menor.*

Demostración:

$\Rightarrow$ Ser primitivo salvo constante significa en particular que f es irreducible, por lo que su orden es el de cualquiera de sus raíces. Como una de las raíces de f es un elemento primitivo de $\mathbb{F}_{q^n}$, $\mathrm{ord}(f) = q^n - 1$ (en particular, todas las raíces de f son primitivas, un hecho que no comentamos antes, pero se podría haber deducido de la Proposición A.27).

$\Leftarrow$ Ahora solo debemos probar que f es irreducible: en ese caso el orden de cualquiera de sus raíces será $q^n - 1$, y en consecuencia f será primitivo.
Veamos que si f no fuera irreducible, $\mathrm{ord}(f) < q^n - 1$. Supongamos que f es reducible, o bien $f = g^b$ con $b \geq 2$ y g irreducible, o bien $f = g_1 g_2$ con g_1, g_2 coprimos.
En el primer caso sea $m = n/b$ el grado de g, sea $c = \mathrm{ord}(g)$. Sea t el menor natural con $p^t \geq b$, donde p es la característica de $\mathbb{F}_q$. Tenemos

$$\mathrm{ord}(f) = ep^t \leq (q^m - 1)p^t < q^{m+t} - 1 \tag{1.1}$$

pero teníamos $ep^{t-1} < b$, lo que nos da la segunda desigualdad en la siguiente cadena

$$t \leq p^{t-1} \leq b - 1 \leq (b-1)m,$$

y terminamos introduciendo esta última relación $t \leq (b-1)m$ en los exponentes de 1.1 se obtiene

$$\mathrm{ord}(f) < q^{m+t} - 1 \leq q^{bm} - 1 = q^n - 1.$$

En el segundo caso, sea m_i el grado y c_i el orden de cada polinomio g_1, g_2. Como $g_i \mid (x^{c_i} - 1)$ y $(x^{c_i} - 1) \mid (x^{c_1 c_2} - 1)$, y $\gcd(g_1, g_2) = 1$, debe ocurrir que $g_1 g_2 \mid (x^{c_1 c_2} - 1)$, y en consecuencia tenemos

$$\mathrm{ord}(f) \leq c_1 c_2 \leq (q^{m_1} - 1)(q^{m_2} - 1) < q^{m_1 + m_2} - 1 = q^n - 1.$$

Con esto termina la demostración.

□

Ya hemos comentado qué son los polinomios primitivos, y hemos probado algunas propiedades. Concluimos esta sección esbozando resumidamente cómo se encuentran estos polinomios primitivos. En general, no hay mucho nuevo que decir:

Observación 1.17. Para buscar polinomios primitivos en $\mathbb{F}_q[x]$, se buscan primero polinomios irreducibles. Una vez que tenemos un candidato f irreducible de grado n, sabemos que es primitivo si su orden es maximal, es decir, $q^n - 1$. En todo caso, el orden debe dividir a $q^n - 1$, si tenemos la factorización de este número, podremos probar solo con los factores propios t de

$q^n - 1$: si para ninguno $f \mid x^t - 1$, entonces f es primitivo. En particular, si $q = 2$, y $2^n - 1$ es un primo de Mersenne, todos los polinomios irreducibles en $\mathbb{F}_2[x]$ con grado n son primitivos.

1.2. Caracteres de un grupo

La Teoría de Caracteres permite ver un grupo abeliano G en el familiar contexto de los números complejos. Daremos una breve introducción con los resultados más elementales.

Definición 1.18. Un carácter (complejo) sobre un grupo abeliano G es un homomorfismo de grupos $\chi\colon G \to \mathbb{C}^\times$ de G al grupo multiplicativo $\mathbb{C}^\times$ del cuerpo de los números complejos $\mathbb{C}$.
Si G es un grupo finito, para todo $g \in G$ existe $m \in \mathbb{N}$ tal que $mg = 1$, por lo que se debe verificar $|\chi(g)| = 1$ para cualquier carácter χ. En consecuencia, cuando tratemos con grupos finitos, se puede considerar que las imagenes de los caracteres están restringidas a $\mathbb{T} = \{z \in \mathbb{C}\colon |z| = 1\}$.

Nota 1.19. El grupo $\mathbb{T}$ es de hecho el grupo de Lie $U(1)$. Esto es un vestigio de una definición más general de carácter proveniente de la Teoría de Representación. Dada una representación ρ de un grupo G sobre un cuerpo K, el carácter asociado a esta representación es

$$\chi_\rho(g) = \mathrm{Tr}(\rho(g)).$$

Como vemos, la Definición 1.18, que fue la definición original propuesta por el matemático alemán Ferdinand Georg Frobenius en 1896, es el caso particular para representaciones unidimensionales sobre los complejos de esta definición más general. Esta generalización que data de 1977, se debe al matemático francés Jean-Pierre Serre.
La Teoría de Representación es una herrramienta valiosa para la Teoría de Grupos, una buena introducción para el caso de grupos finitos se puede ver en [11] (en castellano).

Definición 1.20. Dado un grupo abeliano G, denotamos por $\widehat{G}$ al conjunto de caracteres sobre G. Si definimos la operación $(\chi \cdot \psi)(g) = \chi(g)\psi(g)$, $\widehat{G}$ es también un grupo abeliano, es el grupo de caracteres de G. Siguiendo la notación multiplicativa, denotamos por 1 al carácter trivial, que envía todos los elementos de G a 1.

Observación 1.21. Supongamos que el grupo abeliano G es además finito. Si G es un grupo cíclico con $G \cong \mathbb{Z}/(n)$, entonces $\chi(1)$ es una raíz n-ésima de la unidad. En particular, la elección de $\chi(1)$ determina completamente el carácter χ. Si la raíz n-ésima que se toma como $\chi(1)$ es primitiva, el resto de caracteres serán potencias de χ, por lo que también $\widehat{G} \cong \mathbb{Z}/(n)$.
Por otro lado, si G se descompone como producto de grupos, $G \cong G_1 \times G_2 \times \ldots \times G_n$, es rutina probar que $\widehat{G} \cong \widehat{G_1} \times \widehat{G_2} \times \ldots \times \widehat{G_n}$.
A raíz de estas dos observaciones, y recordando que todo grupo abeliano finito se descompone como producto de grupos cíclicos (este es el Teorema de Clasificación de los Grupos Abelianos Finitos, ver [37] para una prueba sucinta), concluimos que todo grupo abeliano finito G es isomorfo a su grupo de caracteres $\widehat{G}$.
Finalmente, si consideramos los elementos de G como caracteres en $\widehat{\widehat{G}}$, de forma natural $\widehat{\widehat{G}} \cong G$. Esto ocurre porque ambos grupos tienen la misma cantidad de elementos, y la estructura de $\widehat{G}$ como producto de grupos cíclicos permite con facilidad comprobar que elementos distintos en G corresponden a caracteres distintos en $\widehat{G}$.
Debemos advertir que las afirmaciones hechas en esta Observación, que hemos justificado para el caso de G finito, en general no son ciertas para G infinito.

La actuación de los caracteres se puede interpretar geométricamente como una ordenación de los elementos del grupo G en el borde $\mathbb{T}$ del disco unidad $\mathbb{D} = \{z\colon |z| \leq 1\}$, que visualizamos como una mesa redonda con apoyo en el 0. La siguiente Proposición nos dice que esta ordenación mantiene la mesa $\mathbb{D}$ en equilibrio.

Proposición 1.22. *Sea G un grupo finito abeliano, para todo $\chi \in \widehat{G}$ se verifica*

$$\sum_{h\in G}\chi(h) = \begin{cases} 0 & si\ \chi \neq 1, \\ |G| & si\ \chi = 1, \end{cases} \tag{1.2}$$

y para todo $g \in G$ se verifica

$$\sum_{\psi\in\widehat{G}}\psi(g) = \begin{cases} 0 & si\ g \neq 0, \\ |G| & si\ g = 0. \end{cases} \tag{1.3}$$

Demostración:
Si χ no es trivial, existe $a \in G$ tal que $\chi(a) \neq 1$. Entonces

$$\chi(a)\sum_{h\in G}\chi(h) = \sum_{h\in G}\chi(ah) = \sum_{h'\in G}\chi(h'),$$

restando la última expresión a la primera obtenemos $(1-\chi(a))\sum_{h\in G}\chi(h)$, por lo que $\sum_{h\in G}\chi(h) = 0$. La prueba de la segunda parte de la Proposición es análoga. De hecho, ni si quiera es necesaria, teniendo en consideración la dualidad de la Observación 1.21. □

Corolario 1.23 (relación de ortogonalidad de Schur)**.** *Sea G un grupo finito abeliano. Si $\psi, \chi \in \widehat{G}$ son dos caracteres distintos, entonces*

$$\sum_{g\in G}\psi(g)\overline{\chi(g)} = 0,$$

y si $g, h \in G$ son dos elementos distintos, entonces

$$\sum_{\chi\in\widehat{G}}\chi(g)\overline{\chi(h)} = 0.$$

Demostración:
La primera ecuación se obtiene aplicando 1.2 a $\psi\chi^{-1}$, la segunda ecuación se obtiene aplicando 1.3 a $g-h$. □

Observación 1.24. Sea G un grupo abeliano finito. En el $\mathbb{C}$-espacio vectorial $\mathbb{C}^G$ se define el producto interno

$$\langle f_1, f_2\rangle = \frac{1}{|G|}\sum_{g\in G} f_1(g)\overline{f_2(g)}.$$

En particular, los caracteres son elementos de norma 1 en dicho espacio vectorial, y el anterior Corolario 1.23 nos dice que son linealmente independientes en $\mathbb{C}^G$ (de hecho, forman una base). De manera dual, en el $\mathbb{C}$-espacio vectorial $(\mathbb{C}^G)^*$ (que es en el fondo el $\mathbb{C}$-módulo libre generado por G, o también se puede ver como $\mathbb{C}^{\widehat{G}}$) se define el producto interno

$$\langle g_1, g_2\rangle = \frac{1}{|\widehat{G}|}\sum_{\chi\in\widehat{G}}\chi(g_1)\overline{\chi(g_2)}$$

haciendo un abuso de notación, ya que en realidad los $g_1, g_2 \in (\mathbb{C}^G)^*$ son los que se aplican a los $\chi \in \widehat{G}$. En particular, los elementos del grupo son elementos de norma 1 en dicho espacio vectorial, y el anterior Corolario 1.23 nos dice que son linealmente independientes en $(\mathbb{C}^G)^*$ (de hecho, forman una base, pero todo esto ya era evidente desde el punto de vista del $\mathbb{C}$-módulo libre).

La transformada de Fourier es una herramienta fundamental en Matemáticas que toma diversas formas dependiendo del contexto. En nuestro caso, la dualidad entre el grupo G y su grupo de caracteres $\widehat{G}$, junto con la relación de ortogonalidad anterior, sugieren la existencia de una "tranformada de Fourier" para caracteres de grupos.

Definición 1.25. Sea G un grupo abeliano, sea $f\colon G \to \mathbb{C}$ una función. La transformada de Fourier de f se define como

$$\begin{aligned} \widehat{f}\colon \widehat{G} &\to \mathbb{C} \\ \chi &\mapsto \sum_{g\in G} f(g)\chi(g). \end{aligned}$$

Observación 1.26. Si G es además un grupo cíclico con $G \cong \mathbb{Z}/(n)$, y teniendo en consideración la Observación 1.21, la transformada de Fourier de f se puede considerar como una función $\mathbb{Z}/(n) \to \mathbb{C}$. Fijada $\zeta = e^{2\pi i/n}$ una raíz n-ésima primitiva de la unidad, si definimos $\chi_1(1) = \zeta$, entonces $\chi_1(k) = \zeta^k$, y cualquier otro carácter no trivial χ_m será una potencia de χ_1, por lo que $\chi_m(k) = \zeta^{mk}$.
En estas condiciones, y abusando de la notación, la transformada de Fourier de f se escribe como

$$\widehat{f}(m) = \sum_{k=0}^{n-1} f(k)e^{2\pi i(km/n)},$$

con $k, m = 0, \dots, n-1$. Esta forma es sin duda mucho más familiar.

Ahora probaremos unos cuantos resultados clásicos sobre la transformada de Fourier en el caso de los caracteres, con lo que terminamos esta sección.

Proposición 1.27 (fórmula de inversión)**.** *Sea G un grupo abeliano, sea $f\colon G \to \mathbb{C}$ una función. Se puede expresar f como una combinación lineal de caracteres:*

$$f(g) = \frac{1}{|G|}\sum_{\chi\in\widehat{G}} \widehat{f}(\chi)\overline{\chi(g)}.$$

Demostración:
Operemos,

$$\begin{aligned} \frac{1}{|G|}\sum_{\chi\in\widehat{G}} \widehat{f}(\chi)\overline{\chi(g)} &= \frac{1}{|G|}\sum_{\chi\in\widehat{G}}\sum_{h\in G} f(h)\chi(h)\overline{\chi(g)} \\ &= \frac{1}{|G|}\sum_{h\in G} f(h)\sum_{\chi\in\widehat{G}} \chi(h)\overline{\chi(g)} = f(g), \end{aligned}$$

donde la última igualdad es consecuencia de la relación de ortogonalidad (Corolario 1.23). □

Proposición 1.28 (fórmula de convolución)**.** *Sea G un grupo abeliano, sean $f_1, f_2\colon G \to \mathbb{C}$ dos funciones. Se define la convolución de f_1 con f_2 como*

$$\begin{aligned} f_1 \star f_2\colon G &\to \mathbb{C} \\ y &\mapsto \sum_{x\in G} f_1(x)f_2(y-x). \end{aligned}$$

Con esta definición se verifica

$$\widehat{f_1 \star f_2} = \widehat{f_1}\cdot\widehat{f_2}.$$

Demostración:
Operemos,

$$\begin{aligned}(\widehat{f_1 \star f_2})(\chi) &= \sum_{y\in G}(f_1 \star f_2)(y)\chi(y)\\ &= \sum_{y\in G}\sum_{x\in G} f_1(x)f_2(y-x)\chi(y)\\ &= \sum_{x\in G} f_1(x)\chi(x)\sum_{y\in G} f_2(y-x)\chi(y-x)\\ &= \left(\sum_{x_1\in G} f_1(x_1)\chi(x_1)\right)\left(\sum_{x_2\in G} f_2(x_2)\chi(x_2)\right) = \widehat{f}_1(\chi)\cdot\widehat{f}_2(\chi).\end{aligned}$$

□

Proposición 1.29 (identidad de Plancherel)**.** *Sea G un grupo abeliano, sean $f_1, f_2\colon G\to\mathbb{C}$ una función. Se verifica*

$$|G|\sum_{g\in G} f_1(g)\overline{f_2(g)} = \sum_{\chi\in\widehat{G}}\widehat{f}_1(\chi)\overline{\widehat{f}_2(\chi)}.$$

Demostración:
Operemos,

$$\begin{aligned}\sum_{\chi\in\widehat{G}}\widehat{f}_1(\chi)\overline{\widehat{f}_2(\chi)} &= \sum_{\chi\in\widehat{G}}\sum_{g\in G}\sum_{h\in G} f_1(g)\chi(g)\overline{f_2(h)}\,\overline{\chi(h)}\\ &= \sum_{g\in G} f_1(g)\sum_{h\in G}\overline{f_2(h)}\sum_{\chi\in\widehat{G}}\chi(g)\overline{\chi(h)} = |G|\sum_{g\in G} f_1(g)\overline{f_2(g)},\end{aligned}$$

donde la última igualdad es consecuencia de la relación de ortogonalidad. □

Corolario 1.30 (identidad de Parseval)**.** *Sea G un grupo abeliano, sea $f\colon G\to\mathbb{C}$ una función. Se verifica*

$$|G|\sum_{g\in G}|f(g)|^2 = \sum_{\chi\in\widehat{G}}|\widehat{f}(\chi)|^2$$

Demostración:
Es consecuencia inmediata de la Proposición anterior. □

Con esto termina nuestra breve incursión en la Teoría de Caracteres. En verdad, gran de los resultados propuestos aquí no serán utilizados durante este Trabajo de Fin de Grado. Aún así, estos resultados tienen interés por si propio, y ayudan a poner en contexto algunas definiciones del siguiente Capítulo.

Capítulo 2

Introducción a la Criptografía

> The multiple human needs and desires that demand privacy among two or more people in the midst of social life must inevitably lead to cryptology wherever men thrive and wherever they write.
>
> *David Kahn*

En este Capítulo se pretende dar una muy breve introducción a la Criptografía, prestando especial atención a los cifrados en flujo simétricos. Estos conceptos motivan el estudio de lo que llamaremos "LFSRs" en el siguiente Capítulo, que constituyen el eje central de este Trabajo de Fin de Grado.
Los libros de referencia para este Capítulo son [10][Capítulo 3] y [28][Capítulos 6-7], aunque en ocasiones usaremos fuentes más modernas para reflejar los últimos desarrollos y tendencias en este campo.

2.1. Criptosistemas y ataques

Con la invención de la escritura en las civilizaciones antiguas, aparece la necesidad de evitar que la información escrita en un mensaje o documento caiga en malas manos. La Criptografía, del griego romanizado *kryptós* "secreto", y *graphein* "escribir", es la disciplina que tradicionalmente ha estudiado como ocultar esta información de un adversario. Desde el clásico cifrado César utilizado por los romanos, pasando por la máquina Enigma utilizada por la Alemania nazi, y hasta el importante criptosistema RSA (Rivest–Shamir–Adleman), la Criptografía ha sufrido muchas revisiones y revoluciones a lo largo de su historia. Con el advenimiento de la era de la información, la importancia de la Criptografía es ahora mayor que nunca. Adicionalmente, la Criptografía se ha expandido para tratar otros temas afines y no menos esenciales, como la autentificación de datos o el anonimato de las comunicaciones.
Nosotros evitaremos entrar en cuestiones históricas, y tampoco prestaremos atención a todos los distintos aspectos de la Criptografía que no sean estrictamente el cifrado de datos (en todo caso, este es el núcleo central de la Criptografía). De manera excepcional, durante este Capítulo, se relaja el estilo de escritura. Si bien se darán definiciones y resultados, no se pretende formalizar todo rigurosamente. Esto entorpecería el desarrollo del texto, y en todo caso, la mayor parte de este formalismo no volvería a ser usado. Solo pretende motivar y poner en perspectiva el resto de este Trabajo de Fin de Grado.

Definición 2.1. Un criptosistema o cifrado es una 5-upla $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ donde

- $\mathcal{P}$ es el conjunto de mensajes en "texto claro", es decir, aquellos que todavía no han sido procesados por el criptosistema,

- $\mathcal{C}$ es el conjunto de mensajes en "texto cifrado", es decir, aquellos que ya han sido procesados por el criptosistema,
- $\mathcal{K}$ es el conjunto de claves que admite el criptosistema,
- $\mathcal{E} = \{E_k \mid k \in \mathcal{K}\}$ es el conjunto de funciones de "cifrado", una para cada clave,
- $\mathcal{D} = \{D_k \mid k \in \mathcal{K}\}$ es el conjunto de funciones de "descifrado", una para cada clave,

de forma que para cada clave $e \in \mathcal{K}$, existe otra clave $d \in \mathcal{K}$ tal que $D_d(E_e(p)) = p$ para todo texto claro $p \in \mathcal{P}$. Abusando de la notación, también se llama criptosistema a cualquier dispositivo o programa que implemente estos cinco elementos.
Normalmente $\mathcal{P} = \mathcal{C} = \mathcal{W}^n$, donde $\mathcal{W}$ es el "alfabeto" en el que se codifica el mensaje, y n es la longitud del texto claro en caracteres, que se cifra en un texto cifrado de igual longitud. A cada elemento de $\mathcal{W}$ se le denomina "letra", aunque estos elementos pueden ser símbolos arbitrarios.
Si bien en la antigüedad se solía custodiar celosamente el diseño de un criptosistema, en la actualidad se asume que este es conocida públicamente, y que los únicos datos ocultos a un adversario son las claves escogidas. Este es el principio de Kerckhoff: la seguridad de un criptosistema no debe apoyarse en que su estructura interna sea secreta.

La operación estándar de un criptosistema es como sigue.

Algoritmo 2.2. Tenemos un emisor, al que llamaremos "Alice", y un receptor, al que llamaremos "Bob", que pretenden comunicarse sin que una tercera persona, llamada "Eve", sea capaz de descubrir el contenido de la comunicación.

1. Bob selecciona (preferiblemente de manera aleatoria) una clave de descifrado $d \in \mathcal{K}$.
2. Bob genera la clave de cifrado $e \in \mathcal{K}$ correspondiente a la clave de descifrado d.
3. Bob transmite la clave de cifrado e a Alice.
4. Alice cifra su mensaje $p \in P$ aplicando E_e.
5. Alice transmite el mensaje cifrado $E_e(p)$ a Bob.
6. Bob aplica la clave de descifrado d y recupera el mensaje original $D_d(E_e(p)) = p$.

Si Eve intercepa el mensaje cifrado $E_e(p)$, pero no conoce la clave de descifrado d, no podrá obtener el mensaje original p.

Cuando se analiza un criptosistema en Criptografía, se usan con frecuencia los términos "fácil" y "difícil". Si bien estos términos son inherentemente subjetivos, vamos a precisar un poco más a que nos referimos con ellos.

Definición 2.3. Cuando se dice que una computación es "fácil", se entiende que el sistema del usuario indicado puede ejecutarla en poco tiempo (normalmente se refiere a milisegundos). Así mismo, cuando se dice que una computación es "difícil", significa que realizarla en un tiempo razonable (normalmente se refiere a varios años) requeriría significativamente más recursos computacionales (varios órdenes de magnitud más) de los disponibles para un potencial adversario. Por supuesto, los tiempos y recursos exactos dependen de la aplicación.

Queremos pues que un criptosistema sea fácil de usar y difícil de comprometer.

Definición 2.4. Un criptosistema es rápido si es "fácil" para Alice y Bob realizar las operaciones en el Algoritmo 2.2. Se presupone que en todo criptosistema razonable es rápido.
Un criptosistema es seguro si, a partir del mensaje cifrado $E_e(p)$ (junto con otros datos a los que es razonable que tenga acceso), le es "difícil" a Eve obtener información sobre el contenido del mensaje original p. El estatus de seguridad de un criptosistema puede debilitarse a lo largo de los años, según se van descubriendo nuevos ataques (ver la Definición 2.7).

Los criptosistemas aceptables son entonces aquellos que son tanto rápidos como seguros. La parte delicada es evaluar correctamente la seguridad, saber cómo de difícil es encontrar la clave e entre todas las que hay en $\mathcal{K}$.

Observación 2.5. En cualquier aplicación real, el conjunto de claves $\mathcal{K}$ es finito. Se suele decir que la clave es de $\log_2(|\mathcal{K}|)$-bits.

Definición 2.6. Una búsqueda "por fuerza bruta" (brute-force search) sobre el conjunto de claves $\mathcal{K}$ finito consiste en calcular exhaustivamente $D_k(E_e(p))$ para cada $k \in \mathcal{K}$, hasta que topemos con la clave d y descifremos el mensaje (se supone que es posible confirmar por inspección cuando hemos descifrado el mensaje; para un ejemplo donde no es el caso, ver la subsección 2.3). Si la clave de un criptosistema es de n-bit, realizar una búsqueda por fuerza bruta requerirá una media de 2^{n-1} intentos (y un máximo de 2^n intentos en el peor de los casos). Esto exige que cualquier criptosistema seguro tenga un conjunto de claves $\mathcal{K}$ suficientemente grande.

Siempre existe la posibilidad de realizar una búsqueda por fuerza bruta. Esto constituye el punto base respecto al cuál se analiza la seguridad de un criptosistema.

Definición 2.7. Un ataque a un criptosistema es un algoritmo o procedimiento que permite obtener la clave de descifrado d en un cantidad media de operaciones menor que una búsqueda por fuerza bruta. Un ataque teórico es uno que no depende de vulnerabilidades en la implementación o en fallos en el protocolo de comunicación subyacente, sino que compromete directamente la estructura matemática del criptosistema. Existen distintas clases de ataques teóricos, dependiendo de a que partes del criptosistema suponemos que tiene acceso Eve.

Definición 2.8. Se dice que un criptosistema ofrece una seguridad de n bits ante cierta clase de ataque teórico si el mejor ataque conocido de dicha clase obtiene la clave del criptosistema en una media de 2^{n-1} intentos.

Observación 2.9. La existencia de un ataque teórico a un criptosistema no significa necesariamente que este criptosistema deje de ser seguro. Muchas veces los ataques constituyen pequeñas mejoras frente a la fuerza bruta, y solo rebajan la seguridad en unos pocos bits. En ese caso, sigue siendo "difícil" para Eve obtener la clave de descifrado e.

Al igual que la Criptografía se dedica al diseño de criptosistemas, el Criptoanálisis es la disciplina que analiza la seguridad de estos. El estudio conjunto de ambas disciplinas recibe el nombre de Criptología. No es de extrañar que se estudien ambos campos en consonancia: la Criptografía evoluciona inevitablemente ligada al descubrimiento de nuevas vulnerabilidades en los criptosistemas en uso, y las lecciones aprendidas se plasman en el diseño de los criptosistemas más modernos. Desde luego, no es posible tener una perspectiva completa en Criptografía si no se conocen los ataques más importantes.

Nota 2.10. No nos es posible describir el extenso catálogo de ataques desarrollados por el Criptoanálisis. El propósito de esta Nota es simplemente exponer una serie de ejemplos importantes, ejemplos que también surgirán más adelante en el desarrollo del texto.

Primero debemos clarificar que la seguridad del criptosistema no es el único parámetro relevante. Incluso si el propio criptosistema es seguro ante todos los ataques teóricos, no siempre se puede decir lo mismo del protocolo de uso de dicho criptosistema. Por ejemplo, se debe evitar cifrar demasiado texto claro con una misma clave. De lo contrario aparecen colisiones en el texto cifrado, y a partir de las cuales se puede recuperar cierta información sobre el texto claro. Esta es la idea detrás del "ataque del cumpleaños" (birthday attack):

- Ataque del cumpleaños: ataque que aprovecha la llamada "paradoja del cumpleaños", que nos dice que dado un conjunto de elementos de n tipos distintos, con igual probabilidad cada tipo, basta tomar unos $\sqrt{(2\log 2)n}$ elementos para tener una certeza $\geq 1/2$ de que dos sean iguales [9].

Otra consideración importante es que, a menudo, el adversario tiene acceso parcial al criptosistema o puede alterar el canal de comunicación. Esta clase de ataques se denominan "ataques activos" (active attacks):

- Ataque de intermediario (man-in-the-middle attack): ataque en el que el adversario se inserta en la comunicación, interceptando el intercambio de claves y mensajes, mientras hace pensar a ambas partes que se están comunicando directamente. Este ataque no se considera teórico, t criptosistema es vulnerable.

- Ataques de texto claro escogido (chosen-plaintext attack): clase de ataques en el que el adversario puede requerir al criptosistema cifrar textos claros arbitrarios. Al contrario que en el caso anterior, estos ataques sí se consideran teóricos, pues esta situación ocurre con relativa frecuencia, y sí se puede mitigar diseñando adecuadamente el criptosistema.

También debemos guardarnos ante los denominados "ataques de canal lateral" (side-channel attacks). En ellos, el adversario observa pasivamente la ejecución del criptosistema los que Eve obtienen información analizando como un hardware concreto ejecuta las operaciones del criptosistema. Quizás el más notable de esta clase de ataques es el

- Ataques de cronometrado (timing attacks): clase de ataques en los que Eve mide los tiempos de ejecución del criptosistema para distintos mensaje, y hace las deducciones oportunas.

Por último, mencionaremos los ataque cuánticos. Tradicionalmente, la seguridad de los criptosistemas se ha analizado desde el punto de vista de los ordenadores clásicos. Pero en las últimas décadas se esta comenzando a materializar la posibilidad de construir ordenadores cuánticos, y con esta nueva forma de computación llegan también nuevos ataques. Si bien todavía quedan años o incluso décadas antes de que alcancen la escala necesaria para amenazar los criptosistemas actuales, debemos tener en cuenta ya esa posibilidad, si queremos que nuestros textos cifrados no puedan ser comprometidos en el futuro.
Los dos ataques cuánticos más importantes son el algoritmo de Grover y el algoritmo de Shor. Explicaremos brevemente lo que implica cada uno, para más detalles ver [29].

- Algoritmo de Grover: algoritmo que dada una función arbitraria $f\colon X \to Y$ y dado un elemento de la imagen $y \in Y$, encuentra con alta probabilidad una preimagen $x \in f^{-1}(y)$ evaluando f $O(\sqrt{N})$ veces. En nuestro caso, la caja negra es el criptosistema: si este tiene una clave de n-bits, el algoritmo de Grover puede descubrirla en $O(2^{n/2})$ intentos, que es una mejora polinómica frente a los $O(2^n)$ intentos de la búsqueda por fuerza bruta.

- Algoritmo de Shor: algoritmo que resuelve tanto el problema de factorización como el problema del logaritmo discreto en tiempo polinomial. Se sospecha que estos dos problemas no están en P, en cuyo caso el algoritmo de Shor constituiría una mejora exponencial frente a cualquier algoritmo clásico.

2.2. Criptosistemas simétricos y asimétricos

Los criptosistemas se clasifican en dos tipos: los criptosistemas simétricos y los criptosistemas asimétricos.

Definición 2.11. Un criptosistema es simétrico si la clave de cifrado e y la clave de descifrado d son iguales. Denotamos a esta clave única por k. Evidentemente, es necesario que Alice y Bob mantengan esta clave en secreto, debe ser compartida solo por un canal seguro.
Un criptosistema es asimétrico si la clave de cifrado e (llamada "clave pública") y la clave de descifrado d (llamada "clave privada") son distintas, y es "difícil" computar la clave privada d a partir de la clave pública e. Ahora sí es posible para Alice enviar un mensaje confidencial a Bob por un canal que Eve pueda observar: incluso si Eve obtiene la clave pública que Bob le envía a Alice, Eve sigue sin poder computar "fácilmente" la clave privada, por lo que no puede descifrar los mensajes de la comunicación.

Observación 2.12. En el caso de los criptosistemas asimétricos, si bien se admite que el canal de comunicación pueda ser espiado por Eve, es importante que Alice se asegure de que la clave pública que ella recibe es efectivamente la de Bob (de lo contrario, Eve podría insertarse en la comunicación entre Alice y Bob y realizar un ataque de intermediario). Con este fin, se plantea una "infraestructura de clave pública": una serie de protocolos, plataformas y agentes que garantizan la comunicación segura a través de un medio (el ejemplo más notable sería Internet).

Algoritmo 2.13. Un ataque de intermediario se desarrolla como sigue:

1. Bob manda su clave pública e_B por el canal comprometido. Bob piensa que Alice recibirá esa clave.
2. Eve intercepta el mensaje de Bob, y manda a Alice una nueva clave pública e_E.
3. Alice recibe la clave pública e_E de Eve. Alice piensa que esta es la clave pública de Bob.
4. Alice cifra su mensaje p_A con la clave pública de Eve.
5. Alice manda su mensaje cifrado $E_{e_E}(p_A)$ por el canal comprometido. Alice piensa que Bob recibirá este mensaje.
6. Eve intercepta el mensaje de Alice, lo descifra con su clave privada d_E, y obtiene el mensaje en texto claro $D_{d_E}(E_{e_E}(p_A)) = p_A$.
7. Eve cifra su propio mensaje p_E con la clave pública de Bob. Este mensaje normalmente es el mensaje original de Alice, pero no necesariamente.
8. Eve manda el mensaje $E_{e_B}(p_E)$ a Bob.
9. Bob recibe el mensaje de Eve, lo descifra con su clave privada d_B, y obtiene el mensaje en texto claro $D_{d_B}(E_{e_B}(p_E)) = p_E$. Bob piensa que este mensaje proviene de Alice.

Observación 2.14. Si bien, como acabamos de ver, los criptosistemas asimétricos siguen siendo vulnerables a algunos ataques, podría parecer que son estrictamente más seguros que los criptosistemas simétricos. Pero deben tenerse en consideración más factores. En particular, los criptosistemas asimétricos están diseñados principalmente para cifrar mensajes pequeños, y además son considerablemente más lentos que los criptosistemas simétricos. Por esa razón se utilizan los primeros para transmitir de manera segura la clave $k \in \mathcal{K}$ elegida para un criptosistema de clave privada, y el resto de la comunicación se realiza con este último.

Existe un motivo por el que los criptosistemas asimétricos son mucho más lentos que los simétricos.

Observación 2.15. Para satisfacer los requerimientos de clave pública y privada, los criptosistemas asimétricos se basan en las llamadas "funciones trampilla". Estas son familias de funciones que se pueden evaluar (aplicar la clave pública) en tiempo polinomial ("fácilmente"), no se pueden obtener preimagenes en tiempo polinomial ("difícilmente"), y existen datos adicionales (la clave privada) asociada a cada función que permite obtener preimagenes en tiempo polinomial ("fácilmente"). Para su utilización en un criptosistema asimétrico, debe ser además fácil obtener una clave privada aleatoria en el conjunto de claves, y también obtener la clave pública asociada a dicha clave privada.
Notablemente, la existencia de funciones trampilla es un problema abierto (ya que trivialmente implicaría P $\neq$ NP), pero sí existen varios candidatos prácticos (RSA y Diffie-Hellman, por citar los dos más importantes) basados en problemas matemáticos en NP que se sospecha que no están en P (la factorización y el logaritmo discreto, respectivamente).
Por otro lado, los cifrados simétricos no sufren esa limitación. Estos se dedican, esencialmente, a remover y mezclar el texto claro hasta que queda irreconocible. Este procedimiento es de diseño libre, solo depende del buen criterio del criptógrafo (que se debe asegurar de que el resultado final no es vulnerable a los ataque ya existentes). Esta flexibilidad permite escoger operaciones rápidas que reduzcan el tiempo de ejecución, en contraste a los cifrados asimétricos, que van ligados rígidamente a las previamente mencionadas funciones trampilla.

Nota 2.16. Los cifrados asimétricos más usados en la actualidad son extremadamente vulnerables a ataques cuánticos. El ejemplo más notable es el criptosistema RSA: la seguridad de este criptosistema depende de la dificultad del problema de factorización, problema que el algoritmo de Shor resuelve en tiempo polinomial. Es necesario pues buscar nuevos cifrados asimétricos, de esto se ocupa la recién surgida "Criptografía Postcuántica".
La Criptografía Postcuántica también busca nuevos cifrados simétricos, aunque con menor prioridad, ya que estos no se hayan visto tan afectados. Si bien siempre es posible realizar un ataque genérico con el algoritmo de Grover, este es simple de contrarrestar, basta duplicar el tamaño de la clave.

2.3. Libretas de un solo uso

Es posible que un criptosistema sea inmune a incluso las búsquedas por fuerza bruta. Esta propiedad se recoge en la siguiente Definición.

Definición 2.17. Un criptosistema se dice que es "incondicionalmente seguro" cuando no es posible obtener información alguna sobre el texto claro p a partir del texto cifrado c. Más específicamente, p y c son vectores independientes cuando se elige la clave de forma uniforme en $\mathcal{K}$.

El ejemplo por excelencia de criptosistema incondicionalmente seguro son las libretas de un solo uso:

Definición 2.18. Sea el texto claro p una sucesión de bits, y como clave k escogemos una sucesión aleatoria de bits con la misma longitud que el texto claro. El criptosistema que se obtiene tomando como texto cifrado el XOR bit a bit del texto claro y la clave, $c = k \oplus p$, se denomina "libreta de un solo uso".

Si, como su nombre indica, se utiliza cada clave una sola vez, es evidente que este criptosistema es invulnerable a cualquier análisis. Bit por bit, $p_i = k_i \oplus c_i$, y al ser k_i aleatorio con $P(k_i =$

$0) = P(k_i = 1) = 1/2$ e independiente del resto, es imposible obtener información alguna sobre p_i a partir de c. La seguridad de las libretas de un solo uso está supeditada únicamente a que la clave se elija de manera verdaderamente aleatoria, y a que esta se mantenga secreta. En esa línea, el matemático americano Claude Shannon probó en 1949 el siguiente resultado.

Teorema 2.19. *Las libretas de un solo uso son incondicionalmente seguras, y cualquier criptosistema incondicionalmente seguro requiere una clave de la misma o mayor longitud que el mensaje que se vaya a cifrar.*

Demostración:
Ver [35]. □

Pese a sus excelentes garantías de seguridad, las libretas de un solo uso no son prácticas en la mayoría de las situaciones. Más allá de las dificultades para generar una gran cantidad de números aleatorios con distribución uniforme, el problema principal de estas libretas reside en que la clave es tan grande como el mensaje. Si conseguimos transmitir la clave privada de forma segura, ¿qué nos habría impedido haber transmitido el mensaje en vez de la clave?

Nota 2.20. Las libretas de un solo uso siguen teniendo utilidad en aplicaciones de máxima seguridad. Por ejemplo, en los años setenta la NSA (National Security Agency) de Estados Unidos generó miles de libretas para uso militar. En estas situaciones se generan las libretas con antelación, y se distribuye una copia física a aquellos agentes u oficiales con los que uno tenga la intención de comunicarse (incluso antes de conocer el contenido de la posible futura comunicación).

2.4. Distinción entre cifrado en bloque y cifrado en flujo

Tradicionalmente, los sistemas de cifrado se clasifican en dos tipos: los cifrados en bloque, y los cifrados en flujo. Existe cierta ambigüedad en esta clasificación, dependiendo del texto. Con el objetivo de aclarar todo lo posible los términos, nosotros estableceremos una doble distinción:

Definición 2.21.

- Desde el punto de vista operacional, un cifrado en bloque siempre cifra un mensaje completo, con longitud fija, mientras que un cifrado en flujo permite cifrar un tráfico continuo de datos letra a letra, según van llegando. Este punto de vista es el relevante para el uso del criptosistema.
- Desde el punto de vista estructural, un cifrado bloque cifra todo el mensaje simultáneamente, aplicando varias rondas de mezclado que dependen solo de la clave, mientras que un cifrado en flujo mantiene un estado interno que va actualizando, cifrando cada nueva letra dependiendo del estado interno en ese momento. Este punto de vista es el relevante para el análisis del criptosistema.

En resumen, si el criptosistema es una caja negra, la primera clasificación se preocupa por lo que hace la caja, mientras que la segunda se pregunta qué hay dentro de la caja. Normalmente ambas clasificaciones coinciden, con ciertas puntualizaciones y excepciones. Por poner un ejemplo, consideremos el criptosistema Chacha20 [6], que es uno de los dos cifrados simétricos en el protocolo TLS (capa de seguridad del importante protocolo HTTPS). Este criptosistema se suele clasificar como cifrado en flujo (y de hecho lo es desde el punto de vista operacional), pero su funcionamiento interno es análogo al de un cifrado en bloque.

Nota 2.22. Si bien existe criposistemas asimétricos en flujo, la mayoría de los cifrados asimétricos son cifrados en bloque. Esto hace que frecuentemente los criptosistemas asimétricos reciban su propia categoría, y que se entienda que nos referimos a los criptosistemas simétricos cuando hablamos de "cifrado en bloque" o "cifrado en flujo". Por evitar recargar la notación, nosotros adoptaremos este convenio.

Ahora pasaremos a comentar en más detalle, desde el punto de vista estructural, las cuestiones esenciales relativas a cada uno de estos sistemas de cifrado.

2.5. Fundamentos del cifrado en bloque

Detrás del cifrado en bloque está la idea de las permutaciones pseudoaleatorias.

Observación 2.23. Al cifrar un mensaje $p \in \mathcal{W}^n$, el objetivo es que vaya a parar a un elemento aleatorio de $\mathcal{W}^n$. Esto significa que a partir de la clave aleatoria k, queremos tomar al azar una permutación del grupo simétrico $Sym(\mathcal{W}^n)$, que será la que después apliquemos a nuestro mensaje p. Desafortunadamente, el orden del grupo simétrico es $|Sym(\mathcal{W}^n)| = (|\mathcal{W}|^n)!$, lo que significa que es demasiado grande como para codificarlo en una clave k razonable. Por ejemplo, si tratamos con bloques de 64 bits, es fácil comprobar que necesitaríamos una clave de tamaño algo mayor que $1,15$ petabits, absolutamente impráctico. En consecuencia, simplemente se aspira a que, a partir de la clave k elegida al azar, se obtenga una permutación pseudoaleatoria (es decir, que "parezca elegida al azar") de $Sym(\mathcal{W}^n)$.

Otro consideración esencial cuando se diseña un cifrado en bloque es elegir adecuadamente el tamaño del bloque.

Observación 2.24. El bloque debe ser lo suficientemente grande como evitar ciertos ataques a los modos de operación (en particular, se debe evitar el ataque del cumpleaños [27]), pero suficientemente pequeño como para no comprometer ciertos requisitos de velocidad, memoria, rellenado de bloques semillenos, etc. En la actualidad, los tamaños de bloque más comunes son los que se encuentran entre los 64 bits y los 256 bits.

En este sentido, recordemos que un criptosistema de cifrado en bloque solo puede cifrar mensajes del tamaño del bloque. Desde luego, en la mayoría de las aplicaciones es necesario cifrar mensajes de más de 128 bits, acción que no es posible realizar directamente con el cifrado en bloque. Para solventar esto, se emplean los llamados "modos de operación".
El modo más natural de cifrar un mensaje más largo que el tamaño del bloque es el llamado modo EBC.

Definición 2.25. El modo de operación ECB (Electronic CodeBook) divide el mensaje en bloques p_i y aplica la función de cifrado E_k a cada uno de ellos.

Este modo de operación no suele ser recomendable, ya que si dos bloques $p_i = p_j$ son iguales de texto claro, entonces se cifran en bloques iguales $E_k(p_i) = E_k(p_j)$, revelando información sobre el texto claro. Para evitar este problema, otros modos de operación procuran mezclar datos adicionales distintos en el cifrado de cada bloque.

Definición 2.26. Un "vector de inicialización" IV es bloque que se genera al inicio de algunos modos de operación. Para una clave k fija, se exige que los vectores de inicialización sean aleatorios y únicos, pero no necesariamente secretos.

Definición 2.27. El modo de operación OFB (Output FeedBack) cifra repetidas veces el vector de inicialización, y en cada paso lo combina con el bloque de texto claro correspondiente: $o_0 = E_k(IV)$, $o_i = E_k(o_{i-1})$ para $i \geq 1$, $c_i = o_i \oplus p_i$, para todo i.
El modo de operación CFB (Cipher FeedBack) combina cada bloque de texto claro con el bloque de texto cifrado anterior, después de cifrarlo: $c_0 = E_k(IV) \oplus p_0$, $c_i = E_k(c_{i-1}) \oplus p_i$ para $i \geq 1$.
El modo de operación CBC (Cipher Block Chaining) combina cada bloque de texto claro con el bloque de texto cifrado anterior, antes de cifrarlo: $c_0 = E_k(IV \oplus p_0)$, $c_i = E_k(c_{i-1} \oplus p_i)$ para $i \geq 1$.

Nota 2.28. Existen otros muchos modos de operación. En general se utilizan (y son preferibles) los modos que incluyen también la autentificación del mensaje (es decir, la capacidad de comprobar la integridad del mensaje, que no ha sido dañado o modificado por ningún adversario). Aquí cabe nombrar los modos GCM (Galois/Counter Mode) y CCM (Counter with CBC-MAC), que de hecho son los implementados en el protocolo TLS.

2.6. Fundamentos del cifrado en flujo

Al igual que detrás del cifrado en bloque está la idea de permutación pseudoaleatoria, detrás del cifrado en flujo está la idea de sucesión pseudoaleatoria.

Observación 2.29. Los cifrados en flujo generan internamente una sucesión pseudoaleatoria que van combinando con el mensaje. Si esta sucesión es "suficientemente indistinguible" de una sucesión aleatoria real, el cifrado operará como una libreta de un solo uso, que ya sabemos que es un cifrado seguro.

La estructura formal de un cifrado en flujo recoge todos los elementos que contribuyen a la creación de esa sucesión pseudoaleatoria.

Definición 2.30. Un cifrado en flujo mantiene un estado interno σ que en cada paso i se actualiza a σ_i. El mecanismo de actualización puede tomar diversas formas (ver la Definición 2.32), pero es fijo para cada criptosistema. Otra función fija g genera, a partir de la clave k y del estado interno σ_i en cada paso, una letra s_i de la sucesión pseudoaleatoria $\mathbf{s}$ llamada "keystream". Cada letra $s_i = g(k, \sigma_i)$ del keystream se combina con la letra p_i correspondiente del texto claro a través de una tercera función fija h, de modo que $c_i = h(s_i, p_i)$ (en este caso, h no depende de la clave k).
Normalmente las letras se codifican con bits, por lo que se toma como función de combinación un simple XOR bit a bit, $c_i = s_i \oplus p_i$. Los cifrados en flujo con dicha función de combinación se denominan "aditivos".

La elección del XOR bit a bit como función de combinación proviene también de las libretas de un solo uso. En cierto modo resalta el principio de que todo el peso de la seguridad debe recaer sobra la sucesión $\mathbf{s}$.

Nota 2.31. Por motivos de seguridad, para clave k fija, de nuevo es necesario establecer el estado interno σ_0 de un cifrado en flujo a un vector de inicialización IV único. Si la generación del keystream $\mathbf{s}$ no depende del texto claro c, esta unicidad evita un devastador ataque de texto plano: cada letra $c_i = s_i \oplus p_i$, y si con la misma clave k ciframos un texto claro conocido $c'_i = s_i \oplus p'_i$, entonces se obtiene el texto claro original como $p_i = c_i \oplus c'_i \oplus p'_i$. También es importante que el vector de inicialización sea aleatorio, lo que corrige la tendencia de los cifrados en flujo a "tardar en arrancar". es decir, tardar en mezclar el estado interno (y mientras tanto, proporcionan una sucesión pseudoaleatoria de mala calidad). Por último, en casos en los

que ni el mecanismo de actualización del estado interno ni la función g dependan de k, ahora sí es crítico que el vector de inicialización sea secreto, ¡de hecho, se ha convertido en la clave!

El caso descrito al final de la Observación anterior corresponde a uno de los dos tipos principales de cifrado en flujo.

Definición 2.32. Un cifrado en flujo es síncrono si el estado interno depende solo del estado interno anterior, $\sigma_i = f(k, \sigma_{i-1})$ para cierta función fija h.
Un cifrado en flujo es asíncrono si el estado interno depende solo de los últimos t caracteres del texto cifrado, $\sigma_i = (c_{i-t}, \ldots, c_{i-1})$ para $i \geq t$.

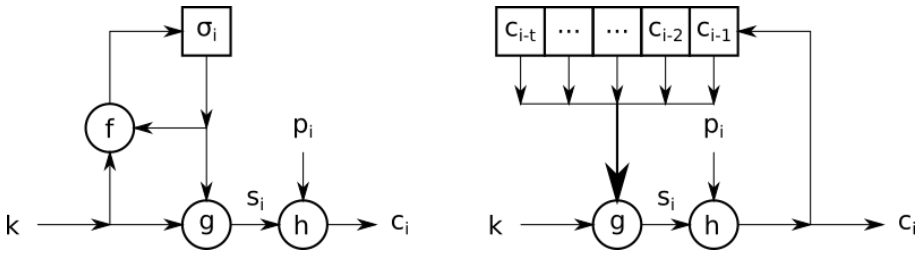

Figura 2.1: Cifrado en flujo síncrono (izquierda), y cifrado en flujo asíncrono (derecha), con la notación de la Definición 2.30.

Nota 2.33. Existen ventajas y desventajas para cada modelo. Recogemos aquí las más importantes.

Cifrados síncronos	Cifrados asíncronos
Sincronización	
El estado interno σ_i debe utilizarse para cifrar la letra p_i. Si se pierde la sincronización entre Alice y Bob (por ejemplo, por la pérdida de algún paquete en la transmisión), el resto del mensaje no se descifra correctamente. Es necesario pues aplicar algún esquema artifical para mantener la sincronización, como introducir marcadores a intervalos regulares en el mensaje cifrado. Por otro lado, al ser el keystream independiente del texto cifrado, es posible pregenerarlo antes de cifrar el texto plano o descifrar el texto cifrado. En algunos casos, es además posible realizar estos procesos con acceso aleatorio, es decir, en comenzando en una posición arbitraria, sin necesidad de cifrar/descifrar todas las letras anteriores, lo que adicionalmente permite paralelizar las operaciones de cifrado o descifrado.	Para descifrar una letra c_i del texto cifrado solo intervienen las t letras anteriores de este, no es necesario mantener ninguna clase de sincronización. Como consecuencia, el acceso aleatorio y la paralelización siempre son posibles, pero solo durante el descifrado.
Propagación de errores	
Un error en la letra p_i del texto claro solo afecta a la letra c_i correspondiente del texto cifrado, y viceversa. Esto significa que los errores no se propagan.	Un error en la letra p_i del texto claro hace indescifrable el resto del texto cifrado más allá de la posición i. Esta propiedad no es siempre negativa: procede de la dispersión de las estadísticas del texto claro, lo que protege a los cifrados síncronos frente a cierto tipo de ataques basado en la posible redundancia de este texto. En contrapartida, un error en la letra c_i del texto cifrado solo hace indescifrables las t letras siguientes.
Ataques activos	
Cualquier intento por un atacante de insertar, borrar, o repetir el texto cifrado causa una pérdida de sincronización, lo cuál se detecta inmediatamente. En contraste, el atacante puede predecir cómo será afectado el texto claro cuando se cambian ciertas letras en el texto cifrado, cambio que puede ser difícil de detectar. Para combatir esto, es necesario emplear mecanismos adicionales autentificación.	Es más difícil detectar algún intento de insertado, borrado, o repetición del texto cifrado, ya que no se produce pérdida de sincronización (pues no existe). También se requiere pues un esquema de autentificación externo. A su vez, cualquier intento de cambiar de letras en el texto cifrado produce t errores en el texto claro, aumentando notablemente las probabilidades de detección.

Por diversas razones, algunas de ellas legítimas, y otras puramente históricas, los criptosistemas de cifrado en flujo no han recibido tanta atención ni gozan de tanta popularidad como los que son cifrados en bloque. Pero no por ello son menos importantes. Analizamos críticamente esta cuestión en la siguiente subsección.

2.7. AES, ¿para qué son necesarios los cifrados en flujo?

> Modern ciphers rarely get broken — at least, not in the Swordfish sense. You're far more likely to get hit by malware or an implementation bug than you are to suffer from a catastrophic attack on AES.
>
> *Matthew Green*

El cifrado en bloque Rijndael surgió como ganador del concurso del NIST (National Institute of Standards and Technology) organizado en 2001 para reemplazar el anticuado y vulnerable DES (Data Encryption Standard). Desde entonces, Rijndael se conoce también por el nombre de AES (Advanced Encryption Standard) [14].
AES ha recibido considerable excrutinio desde su nacimiento. A fecha de hoy existen ataques contra AES, pero ninguno debilita significativamente su nivel de seguridad [8]. La falta de avances significativos en el criptoanálisis de AES desde 2011 se puede considerar como un indicio más de la seguridad de este cifrado en bloque. Estos factores, junto con la velocidad y sencillez de AES, han provocado que sea el cifrado simétrico más empleado en la actualidad. Tal es su popularidad, que Intel y AMD lo han implementado como la extensión AES-NI del conjunto instrucciones x86 en sus CPUs más modernas [18].
AES también es el cifrado *en flujo* más usado en la actualidad. La siguiente Observación explica a qué nos referimos:

Observación 2.34. Los distintos modos de operación en la Definición 2.27 permiten obtener, desde el punto de vista operacional, un cifrado en flujo a partir de un cifrado en bloque. Por ejemplo, si queremos obtener un cifrado en flujo síncrono, basta considerar el modo OFB. Si necesitamos un cifrado en flujo asíncrono, esto se consigue con el modo CFB. Incluso si requerimos la habilidad de cifrar/descifrar con acceso aleatorio, existe otro modo de operación, el modo CTR (CounTeR), que cubre este caso.

En definitiva, no hay nada que haga un cifrado en flujo que no pueda hacer un cifrado en bloque en un modo de operación adecuado. Y estando tan estandarizado el cifrado en bloque AES, cuya seguridad ha sido puesto a prueba por cientos de expertos, la pregunta obvia es ¿para qué preocuparse en desarrollar o analizar otros cifrados en flujo?

Nota 2.35. Existen diversas motivos por los que es beneficioso o prudencial tener alternativas a AES. Veamos:

- Razones de eficiencia:
    - Las CPUs móviles (es decir, las de los móviles), y otros procesadores de bajo consumo (que se están multiplicando con el advenimiento del "internet de las cosas") no implementan AES-NI. En consecuencia, varios cifrados en flujo superan considerablemente a AES (y a otros cifrados en bloque populares) en términos de velocidad, consumo de memoria y/o consumo de área de chip. AES no fue concebido para estos sistemas, en los que se implementa de manera mucho más natural los cifrados en flujo.

- Razones de seguridad:
  - Es muy fácil implementar AES incorrectamente, haciéndolo vulnerable a un ataque de canal lateral. En especial, AES puede ser muy vulnerable a los ataques de temporizado si no se tiene cuidado [7]. Cabe mencionar que las instrucciones de AES-NI en procesadores modernos no se consideran vulnerable a ataques de temporizado.
  - Pero en el caso de AES-NI, surge la cuestión de la confianza en los fabricante de microchips. No sería la primera vez que se implementa una puerta trasera en hardware (basta ver el artículo en Wikipedia https://en.wikipedia.org/wiki/Hardware_backdoor), y ya existe cierta suspicacia hacia componentes como el Intel Management Engine o el AMD Platform Security Processor.
  - Aunque quizás improbable, es posible que algún día se descubra un ataque práctico a AES o se desarrolle algún método de computación capaz de realizar esos ataques. Ante ese hipotético escenario, es importante poseer algoritmos bien aceptados, analizados e implementados para reemplazar AES con la mayor brevedad posibles.

Con esto termina nuestra presentación de la Criptografía. De aquí en adelante, recuperamos el rigor matemático. Nuestro primer cometido será precisar en el siguiente Capítulo que significa que una sucesión pseudoaleatoria sea "suficientemente aleatoria".

Capítulo 3

Sucesiones pseudoaleatorias

> The generation of random numbers is too important to be left to chance.
>
> *Robert Coveyou*

En el Capítulo anterior se exigió que las sucesiones generadas por los cifrados en flujo fueran "pseudoaleatorias". Esta es una propiedad deseable en multitud de aplicaciones, no solo para la Criptografía, y, pese a la popularidad del término, no goza de una definición universalmente aceptada. El objetivo de esta sección es pues explorar y aclarar algunas propiedades que se pueden exigir para que una sucesión sea pseudoaleatoria. Por supuesto, dejaremos muchos otros "tests de pseudoaleatoriedad" sin mencionar siquiera. Una buena referencia para estos es [34]. Este Capítulo se construye sobre la exposición en [16][Capítulos 8-9], con algún resultado adicional, y añadiendo algo de contexto y detalles para facilitar la lectura.

Notación 3.1. Para evitar recargar las definiciones, durante el resto este Trabajo de Fin de Grado denotaremos las sucesiones simplemente por $\mathbf{s}$ en vez de escribir $\{s_n\}_{n=0}^{\infty}$. Salvo que se indique lo contrario, supondremos que estas sucesiones están formadas por elementos de un alfabeto finito $\mathcal{W}$. Cuando haya caracteres involucrados, asumiremos que $\mathcal{W}$ tiene estructura de grupo abeliano. Si se plantean además productos en ese grupo, asumiremos que $\mathcal{W}$ es un cuerpo finito.
Adicionalmente, durante este Capítulo, T siempre denotará el periodo cuando tratemos con una sucesión periódica (en algunos casos se utilizará T incluso si no tratamos con sucesiones periódicas).

Definición 3.2. Una sucesión $\mathbf{s}$ es periódica si existe algún número natural T tal que $s_{n+T} = s_n$ para todo $n \geq 0$. En ese caso, se dice que $\mathbf{s}$ tiene periodo T. Por otro lado, *el* periodo de $\mathbf{s}$ es el menor T que cumple la condición, y por tanto dividirá a cualquier otro T' que la cumpla.

En la siguiente Definición no formalizaremos todavía el concepto de sucesión pseudoaleatoria. De momento mantendremos el término con un significado vago, pues no conviene entrar en detalles prematuramente.

Definición 3.3. Una sucesión aleatoria $\mathbf{s}$ es una sucesión de variables aleatorias independientes uniformemente distribuidas en $\mathcal{W}$.
Una sucesión pseudoaleatoria $\mathbf{s}$ es una sucesión que "se parece suficientemente a un resultado ordinario de una sucesión aleatoria". Las sucesiones pseudoaleatorias suponemos que están generadas por un cifrado en flujo síncrono (o en un lenguaje más matemático, por un "autómata finito"), por lo que deben ser periódicas.

Nota 3.4. Una sucesión aleatoria solo constituye un modelo estadístico contra el que contrastar las sucesiones pseudoaleatorias. Estas sucesiones son generadas de forma determinista, y, por tanto, no son aleatorias. Aún así, en algunos enunciados de esta sección, las sucesiones pseudoaleatoria se considerarán como sucesiones de variables aleatorias con valores constantes en $\mathcal{W}$.

Existen diversas formas de interpretar "parecerse suficientemente". Se puede exigir que no exista un algoritmo que, con probabilidad "estrictamente mayor" que 1/2 y "en tiempo polinómico", distinga la sucesión pseudoaleatoria de una sucesión aleatoria. Esto se relaciona más con Teoría de la Complejidad Computacional que con los LFSRs que veremos más adelante, así que no proseguiremos por este camino.
Otro modo más práctico de proceder es exigir que la sucesión pseudoaleatoria cumpla criterios razonables inspirados por la Combinatoria o la Estadística. Si bien no será posible que se verifiquen *todos* los criterios (el propio mecanismo de generación de la sucesión la distingue de una verdaderamente aleatoria), se pedirá que se cumplan los más importantes, que dependen del uso que se vaya a dar a la sucesión. En las siguientes subsecciones presentaremos los criterios más relevantes para nuestros propósitos, pero antes, reflexionaremos sobre el raciocinio detrás de todos estos criterios.

Observación 3.5. Las sucesiones pseudoaleatorias son periódicas, así que el primer criterio que se debe exigir (y con frecuencia no se especifica debido a su gran obviedad) es que el periodo sea extremadamente grande. Más concretamente, para cualquier aplicación práctica con una sucesión pseudoaleatoria fija, el periodo debe ser tal que nunca lo agotemos, de forma que nunca se observe la teórica periodicidad. Por lo tanto, el periodo T se puede considerar como infinito, y por la Ley de los Grandes Números, es razonable esperar que las distintas variables aleatorias que definiremos más adelante converjan todas a su esperanza.

Observación 3.6. En [16], los resultados involucrando la esperanza se prueban para una sucesión elegida al uniformemente al azar en el conjunto de todas las sucesiones de periodo T (dicho de otro modo, consideramos una T-upla aleatoria que se repite, en vez de una sucesión aleatoria). A nosotros nos pareció más interesante enfocar esta sección desde el punto de vista de las sucesiones aleatorias, pero en realidad los resultados y las pruebas para ambos casos son esencialmente idénticos.

3.1. Bloques y rachas

El criterio más común que se puede pedir a una sucesión pseudoaleatoria es que no existan elementos de $\mathcal{W}$ que predominen sobre el resto. Lo que desarrollaremos en esta sección es más fuerte: si consideramos agrupaciones de elementos, tampoco debe observarse ningún sesgo.

Definición 3.7. Las k-uplas $\mathbf{b} = (b_0, \ldots, b_{k-1})$ de elementos de $\mathcal{W}$ reciben se denominan bloques de longitud k. Una aparición de un bloque $\mathbf{b}$ en una sucesión $\mathbf{s}$ es un índice i tal que $s_i = b_0, s_{i+1} = b_1, \ldots, s_{i+k-1} = b_{k-1}$.
Un bloque con k elementos iguales $s_i = s_{i+1} = \ldots = s_{i+k-1}$, y tal que los elementos extremos sean distintos, $s_{i-1} \neq s_i$ (o $i = 0$) y $s_{i+k-1} \neq s_{i+k}$, se denomina racha de longitud k.

Proposición 3.8. *Sea* $\mathbf{b}$ *un bloque de longitud* k *y sea* $\mathbf{s}$ *una sucesión de variables aleatorias. Fijado un periodo* T*, definimos la variable aleatoria*

$$N_{\mathbf{b},T}(\mathbf{s}) = |\{0 \leq i < T \mid i \text{ es una aparición de } \mathbf{b}\}|.$$

Si $\mathbf{s}$ *es una sucesión aleatoria, se verifica*

$$E[N_{\mathbf{b},T}(\mathbf{s})] = \frac{T}{|\mathcal{W}|^k}.$$

Demostración:
Nos fijamos en los índices $0, 1, \ldots, T+k$. Para cada índice i con $0 \leq i < T$, si fijamos una aparición del bloque $\mathbf{b}$ en i, existen $|\mathcal{W}|^T$ formas de rellenar los T índices restantes. Deducimos que en total hay $T|\mathcal{W}|^T$ posibles apariciones de $\mathbf{b}$, y puesto que todas las elecciones de los $T+k$ índices son equiprobables, basta dividir para calcular la esperanza:

$$E[N_{\mathbf{b},T}(\mathbf{s})] = \frac{T|\mathcal{W}|^T}{|\mathcal{W}|^{T+k}} = \frac{T}{|\mathcal{W}|^k}.$$

□

Definición 3.9. Decimos que una sucesión pseudoaleatoria $\mathbf{s}$ está equidistribuida hasta orden k si para cada cada m con $1 \leq m \leq k$, y para cada bloque $\mathbf{b}$ de longitud m, el número de bloques $N_{\mathbf{b},T}(\mathbf{s})$ satisface

$$\left\lfloor \frac{T}{|\mathcal{W}|^m} \right\rfloor \leq N_{\mathbf{b},T}(\mathbf{s}) \leq \left\lceil \frac{T}{|\mathcal{W}|^m} \right\rceil.$$

Si $\mathbf{s}$ está equidistribuida hasta orden 1, también se dice que $\mathbf{s}$ está equilibrada.

Ahora daremos un resultado y una definición análogos para las rachas.

Proposición 3.10. *Sea* $\mathbf{s}$ *una sucesión de variables aleatorias. Fijado un periodo* T*, definimos la variable aleatoria*

$$R_{k,T}(\mathbf{s}) = |\{1 \leq i < T+1 \mid i \text{ es una aparición de una racha de longitud } k\}|.$$

Si $\mathbf{s}$ *es una sucesión aleatoria, se verifica*

$$E[R_{k,T}(\mathbf{s})] = \frac{T(|\mathcal{W}|-1)^2}{|\mathcal{W}|^{k+1}}.$$

Demostración:
Nos fijamos en los índices $0, 1, \ldots, T+k+1$. Para cada índice i con $1 \leq i < T+1$, existen $|\mathcal{W}|$ elecciones del elemento que se repite en la racha, $|\mathcal{W}|-1$ elecciones para cada elemento que limita la racha en un extremo, y $|\mathcal{W}|^{T-1}$ formas de rellenar los $T-2$ índices restantes. Entonces existen $T|\mathcal{W}|^T(|\mathcal{W}|-1)^2$ posibles apariciones de una racha de longitud k, y puesto que todas las elecciones de los $T+k+1$ índices son equiprobables, basta dividir para calcular la esperanza:

$$E[N_{\mathbf{b},T}(\mathbf{s})] = \frac{T|\mathcal{W}|^T(|\mathcal{W}|-1)^2}{|\mathcal{W}|^{T+k+1}} = \frac{T(|\mathcal{W}|-1)^2}{|\mathcal{W}|^{k+1}}.$$

□

Definición 3.11. Decimos que una sucesión pseudoaleatoria $\mathbf{s}$ cumple la propiedad de las rachas si para todo k, el número $R_{k,T}(\mathbf{s})$ de rachas de longitud k satisface

$$\left\lfloor \frac{T(|\mathcal{W}|-1)^2}{|\mathcal{W}|^{k+1}} \right\rfloor \leq R_{k,T}(\mathbf{s}) \leq \left\lceil \frac{T(|\mathcal{W}|-1)^2}{|\mathcal{W}|^{k+1}} \right\rceil.$$

3.2. Desequilibro respecto a un carácter

El siguiente criterio está estrechamente relacionado con la imagen mental de la mesa que planteamos antes de la Proposición 1.22.
También puede ser interesante vincularlo con el criterio de Weyl [21]. En ese sentido, proponemos el siguiente resultado.

Proposición 3.12. *Sea* $\mathbf{s}$ *una sucesión aleatoria y sea* χ *un carácter no trivial. Se verifica:*

$$\frac{1}{T}\sum_{i=0}^{T-1}\chi(s_i) \xrightarrow[T\to\infty]{a.s.} 0.$$

Demostración:
Es consecuencia inmediata de la Proposición 1.22 y de la Ley Fuerte de los Grandes Números. □

Definición 3.13. Sea $\mathbf{s}$ una sucesión periódica y sea χ un carácter no trivial. El desequilibrio de $\mathbf{s}$ respecto a χ es el número complejo

$$Z_\chi(\mathbf{s}) = \sum_{i=0}^{T-1}\chi(s_i).$$

Se dice que una sucesión pseudoaleatoria $\mathbf{s}$ está equilibrada respecto a χ si $|Z_\chi(\mathbf{s})| \leq 1$.

Esta nueva noción de "sucesión equilibrada" difiere de la definida en la subsección anterior. Pero la coincidencia de nombres no es fruto de una elección cuestionable de notación. Por el contrario, refleja la estrecha relación existente entre los dos conceptos.

Proposición 3.14. *Sea* $\mathbf{s}$ *una sucesión periódica. Se verifican:*

1. *Si* $\mathbf{s}$ *está equilibrada con respecto a cualquier carácter no trivial* χ*, entonces* $\mathbf{s}$ *está equilibrada.*

2. *Si* $\mathbf{s}$ *está equilibrada y además*

$$T \equiv a \bmod |\mathcal{W}|, \text{ con } a = -1,\ 0 \ o\ 1$$

entonces $\mathbf{s}$ *está equilibrada con respecto a cualquier carácter no trivial* χ*.*

Demostración:

1. Sea $\mu_{\mathbf{s}}(g)$ el número de veces que $g \in \mathcal{W}$ aparece en un periodo de $\mathbf{s}$. Con esta notación,

$$Z_\chi(\mathbf{s}) = \sum_{i=0}^{T-1}\chi(s_i) = \sum_{g\in\mathcal{W}}\mu_{\mathbf{s}}(g)\chi(g) = \widehat{\mu_{\mathbf{s}}}(\chi).$$

Aplicando la fórmula de inversión de la transformada de Fourier (Proposición 1.27), tenemos

$$\mu_{\mathbf{s}}(g) = \frac{1}{|\mathcal{W}|}\left(T + \sum_{\substack{\chi\in\widehat{\mathcal{W}}\\ \chi\neq 1}}\widehat{\mu_{\mathbf{s}}}(\chi)\chi(g)\right).$$

Finalmente, como $\mathbf{s}$ está equilibrada respecto a cualquier carácter no trivial,

$$\left|\sum_{\substack{\chi\in\widehat{\mathcal{W}}\\ \chi\neq 1}} \widehat{\mu_{\mathbf{s}}}(\chi)\chi(g)\right| \leq |\mathcal{W}| - 1,$$

y en consecuencia $\mu(g) = \lfloor T/|\mathcal{W}|\rfloor$ o $\mu(g) = \lceil T/|\mathcal{W}|\rceil$ para todo $g \in \mathcal{W}$.

2. Dependiendo del valor de a, existe un elemento $g_0 \in \mathcal{W}$ que aparece una vez menos, igual de veces, o una vez más que el resto. Entonces consideramos

$$Z_\chi(\mathbf{s}) = \sum_{i=0}^{T-1} \chi(s_i) = \sum_{g\in\mathcal{W}} \mu_{\mathbf{s}}(g)\chi(g) = \lfloor T/|\mathcal{W}|\rfloor \sum_{g\in\mathcal{W}} \chi(g) + a\chi(g_0),$$

y recordando la Proposición 1.22, obtenemos

$$|Z_\chi(\mathbf{s})| = |0 + a\chi(g_0)| \leq 1.$$

□

3.3. Autocorrelación

Nuestras sucesiones pseudoaleatorias son inherentemente periódicas, pero ya comentamos que este periodicidad nunca se llega a observar. Esto no impide que existe una subestructura periódica con periodo mucho menor. La autocorrelación es una herramienta excelente que nos permite detectar si efectivamente existen esas subestructuras.

Definición 3.15. Sea $\mathbf{s}$ una sucesión, sea $\tau \geq 0$ un entero no negativo. Se define la sucesión desplazada $\mathbf{s}^\tau$ de $\mathbf{s}$ como $s_n^\tau = s_{n+\tau}$ para todo $n \geq 0$, y τ recibe el nombre de desplazamiento.

Definición 3.16. Fijado un periodo T y un carácter no trivial χ, se define la función de correlación cruzada entre dos sucesiones $\mathbf{a}$ y $\mathbf{b}$ como

$$\mathcal{C}_{\mathbf{a},\mathbf{b}}(\tau) = \sum_{i=0}^{T-1} \chi(a_i)\overline{\chi(b_i^\tau)} = \sum_{i=0}^{T-1} \chi(a_i)\overline{\chi(b_{i+\tau})}.$$

La función de autocorrelación de una sucesión $\mathbf{s}$ se define como $\mathcal{A}_{\mathbf{s}}(\tau) = \mathcal{C}_{\mathbf{s},\mathbf{s}}(\tau)$. Es la correlación cruzada de la sucesión $\mathbf{s}$ consigo misma.

Notación 3.17. Por supuesto, sería apropiado que en los símbolos "$\mathcal{C}_{\mathbf{a},\mathbf{b}}(\tau)$", "$\mathcal{A}_{\mathbf{s}}(\tau)$" viniera incluido el periodo T y el carácter no trivial χ fijados . No se ha hecho esto por evitar recargar innecesariamente la notación. Para los resultados que trataremos, el periodo será o bien irrelevante (para sucesiones aleatorias), o bien el de la sucesión (para sucesiones pseudoaleatorias). Por otro lado, la elección el carácter no trivial no jugará ningún papel.

Cabe mencionar que la prueba de la siguiente Demostración se puede realizar sin referencia al periodo, solamente explotando las propiedades de simetría de los caracteres de grupos finitos abelianos. Nosotros, siguiendo la filosofía de las otras pruebas de esta sección, nos decantaremos por un enfoque más combinatorio.

Proposición 3.18. *Si* $\mathbf{s}$ *es una sucesión de variables aleatorias,* $A_{\mathbf{s}}(\tau)$ *se puede considerar como una variable aleatoria.*
Si $\mathbf{s}$ *es una sucesión aleatoria, se verifica:*

$$E[\mathcal{A}_{\mathbf{s}}(\tau)] = \begin{cases} T & si\ \tau = 0, \\ 0 & si\ \tau \neq 0, \end{cases}$$

Demostración:
El caso $\tau = 0$ es trivial. Si $\tau \neq 0$, fijamos un periodo T, y denotamos por $\mathcal{S}$ el conjunto de las $|\mathcal{W}|^T$ posibles elecciones de los elementos con índices $0, 1, \ldots, T-1$. El valor esperado de $\mathcal{A}_{\mathbf{s}}(\tau)$ en este caso es

$$E[\mathcal{A}_{\mathbf{s}}(\tau)] = \frac{1}{|\mathcal{W}|^T} \sum_{\mathbf{s}\in\mathcal{S}} \sum_{i=0}^{T-1} \chi(s_i)\overline{\chi(s_{i+\tau})} = \frac{1}{|\mathcal{W}|^T} \sum_{i=0}^{T-1} \sum_{\mathbf{s}\in\mathcal{S}} \chi(s_i)\chi(-s_{i+\tau})$$

Para cada $g, h \in \mathcal{W}$, y para cada $0 \leq i < T$, hay $|\mathcal{W}|^{T-2}$ elecciones $\mathbf{s} \in \mathcal{S}$ tal que $s_i = g$ y $s_{i+\tau} = h$. Esto permite reescribir la suma interior como:

$$E[\mathcal{A}_{\mathbf{s}}(\tau)] = \frac{1}{|\mathcal{W}|^T} \sum_{i=0}^{T-1} |\mathcal{W}|^{T-2} \sum_{g\in\mathcal{W}} \sum_{h\in\mathcal{W}} \chi(g)\chi(-h) = \frac{1}{|\mathcal{W}|^2} \sum_{i=0}^{T-1} \left(\sum_{g\in\mathcal{W}} \chi(g) \right)^2 = 0,$$

la última igualdad por la Proposición 1.22. □

Definición 3.19. Se dice que una sucesión pseudoaleatoria $\mathbf{s}$ tiene función de autocorrelación ideal si para cualquier carácter no trivial χ fijo, y para todo desplazamiento no nulo τ, se cumple

$$|\mathcal{A}_{\mathbf{s}}(\tau)| \leq 1.$$

3.4. Sucesiones de de Bruijn

La siguiente definición se debe al matemático holandés Nicolaas Govert de Bruijn. No es una errata que aparezca la palabra "de" dos veces seguidas.

Definición 3.20. Una sucesión de de Bruijn de orden k es una sucesión periódica $\mathbf{s}$ tal que cada bloque de longitud k aparece exactamente una vez en cada periodo.
Si marcamos un elemento $w^* \in \mathcal{W}$, una sucesión agujereada de de Bruijn de orden k es una sucesión periódica $\mathbf{s}$ tal que cada bloque de longitud k, excepto el bloque $w^*w^* \ldots w^*$ (de longitud k), aparece exactamente una vez en cada periodo. Normalmente $\mathcal{W}$ tiene estructura de grupo, y el elemento marcado es el 0.
Se supone que $|\mathcal{W}| \geq 1$ para las sucesiones de de Bruijn, $|\mathcal{W}| \geq 2$ para las sucesiones de de Bruijn agujereadas, y por supuesto $k \geq 1$ en ambos casos. Esto es importante para algunas desigualdades estrictas que se plantearán después.
Notemos que quitando o añadiendo un w^* al bloque $w^*w^* \ldots w^*$, obtenemos una correspondencia biyectiva entre el conjunto de sucesiones de de Bruijn de orden k y el conjunto de sucesiones de de Bruijn agujereadas de orden k.

Proposición 3.21. *Las sucesiones* $\mathbf{s}$ *que son de de Bruijn de orden* k *verifican las siguientes propiedades:*

1. *El periodo de* $\mathbf{s}$ *es* $T = |\mathcal{W}|^k$.

2. *En un periodo de* $\mathbf{s}$ *aparece* $|\mathcal{W}|^{k-m}$ *veces cualquier bloque* $\mathbf{b}$ *de longitud* $m \leq k$. *En consecuencia,* $\mathbf{s}$ *está equidistribuida hasta orden* k.

3. *En un periodo de* $\mathbf{s}$ *aparecen* $|\mathcal{W}|^{k-m-1}(|\mathcal{W}|-1)^2$ *rachas de longitud* m *para todo* $m \leq k-2$, *aparecen* $|\mathcal{W}|(|\mathcal{W}|-2)$ *rachas de longitud* $k-1$, *aparecen* $|\mathcal{W}|$ *rachas de longitud* k, *y no aparecen rachas con longitud mayor que* k.

Demostración:

1. Cada aparición de un bloque en un periodo es un índice del periodo, y todo índice corresponde a un bloque.

2. Fijado un bloque de longitud m, existen $|\mathcal{W}|^{k-m}$ formas de rellenarlo a un bloque de longitud k, y cada bloque de longitud k aparece a lo sumo una vez. Para la segunda parte, basta notar que

$$\left\lfloor \frac{|W|^k}{|W|^m} \right\rfloor = |W|^{k-m} = \left\lceil \frac{|W|^k}{|W|^m} \right\rceil.$$

3. Para $m \leq k-2$, existen $|\mathcal{W}|$ formas de elegir que elemento constituye la racha, y $(|\mathcal{W}|-1)^2$ de elegir los extremos. Se termina con el apartado anterior: el bloque correspondiente a la racha, junto con sus dos extremos, aparece $|\mathcal{W}|^{k-m-2}$ veces en un periodo. Para $m = k-1$, esta vez solo debemos elegir el extremo izquierdo para completar el bloque de longitud k. De los $|\mathcal{W}|-1$ elementos disponibles, uno es extremo izquierdo de la racha de longitud k del elemento que habíamos considerado para la racha de longitud $k-1$. Por tanto, tampoco podemos escogerle, de ahí el $|\mathcal{W}|-2$. Para $m = k$, existe exactamente un bloque correspondiente a cada racha. No puede existir ninguna racha de longitud mayor que k, pues de lo contrario habría dos bloques idénticos consecutivos dentro de esa misma racha.

□

Las sucesiones de de Bruijn están muy cerca de cumplir la propiedad de las rachas. Veamos que las sucesiones de de Bruijn agujereadas sí la cumplen.

Corolario 3.22. *Las sucesiones* $\mathbf{s}$ *que son de de Bruijn agujeradas de orden* k *verifican las siguientes propiedades:*

1. *El periodo de* $\mathbf{s}$ *es* $T = |\mathcal{W}|^k - 1$.

2. *En un periodo de* $\mathbf{s}$ *aparece* $|\mathcal{W}|^{k-m}$ *veces cualquier bloque de longitud* $m \leq k$, *excepto los bloques* $w^*w^*\dots w^*$ *de longitud* m, *que aparecen* $|\mathcal{W}|^{k-m}-1$ *veces. En consecuencia,* $\mathbf{s}$ *está equidistribuida hasta orden* k.

3. *En un periodo de* $\mathbf{s}$ *aparecen* $|\mathcal{W}|^{k-m-1}(|\mathcal{W}|-1)^2$ *rachas de longitud* m *para todo* $m \leq k-2$, *aparecen* $|\mathcal{W}|(|\mathcal{W}|-2)+1$ *rachas de longitud* $k-1$, *aparecen* $|\mathcal{W}|-1$ *rachas de longitud* k, *y no aparecen rachas con longitud mayor que* k. *En consecuencia,* $\mathbf{s}$ *cumple la propiedad de las rachas.*

Demostración:

1. Consecuencia inmediata de la Proposición anterior y de la correspondencia biyectiva de la Definición 3.20.

2. La primera parte es consecuencia inmediata de la Proposición anterior y de la corres-

pondencia biyectiva de la Definición 3.20. Para ver que $\mathbf{s}$ está equidistribuida, basta notar que

$$|W|^{k-m} - 1 = \frac{|W|^k - |W|^m}{|W|^m} < \frac{|W|^k - 1}{|W|^m} < \frac{|W|^k}{|W|^m} = |W|^{k-m}$$

por tanto

$$\left\lfloor \frac{|W|^k - 1}{|W|^m} \right\rfloor = |W|^{k-m} - 1,$$
$$\left\lceil \frac{|W|^k - 1}{|W|^m} \right\rceil = |W|^{k-m}.$$

3. La primera parte es consecuencia inmediata de la Proposición anterior y de la correspondencia biyectiva de la Definición 3.20. Para ver que $\mathbf{s}$ cumple la propiedad de las rachas, observamos que si $m \leq k-1$ tenemos

$$\begin{aligned} |\mathcal{W}|^{k-m-1}(|\mathcal{W}|-1)^2 - 1 &= \frac{|\mathcal{W}|^k(|\mathcal{W}|-1)^2 - |\mathcal{W}|^{m+1}}{|\mathcal{W}|^{m+1}} \\ &< \frac{(|\mathcal{W}|^k - 1)(|\mathcal{W}|-1)^2}{|\mathcal{W}|^{m+1}} \\ &< \frac{|\mathcal{W}|^k(|\mathcal{W}|-1)^2}{|\mathcal{W}|^{m+1}} = |\mathcal{W}|^{k-m-1}(|\mathcal{W}|-1)^2, \end{aligned}$$

por tanto

$$\left\lfloor \frac{(|\mathcal{W}|^k - 1)(|\mathcal{W}|-1)^2}{|\mathcal{W}|^{m+1}} \right\rfloor = |\mathcal{W}|^{k-m-1}(|\mathcal{W}|-1)^2 - 1,$$
$$\left\lceil \frac{(|\mathcal{W}|^k - 1)(|\mathcal{W}|-1)^2}{|\mathcal{W}|^{m+1}} \right\rceil = |\mathcal{W}|^{k-m-1}(|\mathcal{W}|-1)^2.$$

Por otro lado, si $m = k$ tenemos,

$$|\mathcal{W}| - 2 = \frac{|\mathcal{W}|^{k+1}(|\mathcal{W}|-2)}{|\mathcal{W}|^{k+1}} < \frac{(|\mathcal{W}|^k - 1)(|\mathcal{W}|-1)^2}{|\mathcal{W}|^{k+1}} < \frac{|\mathcal{W}|^{k+1}(|\mathcal{W}|-1)}{|\mathcal{W}|^{k+1}} = |\mathcal{W}| - 1,$$

la primera desigualdad porque $|\mathcal{W}|^k - |\mathcal{W}|^2 + 2|\mathcal{W}| - 1 > 0$, y por tanto

$$\left\lfloor \frac{(|\mathcal{W}|^k - 1)(|\mathcal{W}|-1)^2}{|\mathcal{W}|^{k+1}} \right\rfloor = |\mathcal{W}| - 2,$$
$$\left\lceil \frac{(|\mathcal{W}|^k - 1)(|\mathcal{W}|-1)^2}{|\mathcal{W}|^{k+1}} \right\rceil = |\mathcal{W}| - 1.$$

Finalmente, si m es mayor que k,

$$\left\lfloor \frac{(|\mathcal{W}|^k - 1)(|\mathcal{W}| - 1)^2}{|\mathcal{W}|^{m+1}} \right\rfloor = 0,$$
$$\left\lceil \frac{(|\mathcal{W}|^k - 1)(|\mathcal{W}| - 1)^2}{|\mathcal{W}|^{m+1}} \right\rceil = 1.$$

□

Terminamos con un resultado profundo:

Teorema 3.23. *El número de sucesiones de de Bruijn de orden k, y el número de sucesiones de de Bruijn agujereadas de orden k, es*

$$(|\mathcal{W}|!)^{|\mathcal{W}|^{k-1}} |\mathcal{W}|^{-k}.$$

Como consecuencia inmediata de la correspondencia biyectiva de la Definición 3.20, este también es el número de sucesiones de de Bruijn agujereadas de orden k.

Demostración:
Ver [32]. □

3.5. Propiedad de desplazamiento y suma

Al igual que en la subsección anterior ser de de Bruijn implicaba una buena distribución de bloques y rachas, ahora veremos que la "propiedad de desplazamiento y suma" garantiza (bajo ciertas condiciones) una función de autocorrelación ideal.

Definición 3.24. Sea $\mathbf{s}$ una sucesión periódica. Se dice que $\mathbf{s}$ cumple la propiedad de desplazamiento y suma si para cualquier desplazamiento τ se verifica una de las dos siguientes condiciones

1. $\mathbf{s} + \mathbf{s}^\tau$ es la sucesión nula.
2. Existe otro desplazamiento τ' tal que $\mathbf{s} + \mathbf{s}^\tau = \mathbf{s}^{\tau'}$. Como la sucesión es periódica, se puede suponer $0 \leq \tau' < T$.

En la siguiente Proposición veremos que esta propiedad es equivalente a una aparentemente más fuerte. Es interesante también la interpretación vectorial que se da en la Demostración de este resultado.

Proposición 3.25. *Sea $\mathbf{s}$ una sucesión periódica que cumple la propiedad de desplazamiento y suma. Entonces, para cualquier $a, b \in \mathbb{Z}$, y para cualquier desplazamiento τ, se verifica una de las dos siguientes condiciones:*

1. *$a\mathbf{s} + b\mathbf{s}^\tau$ es la sucesión nula.*
2. *Existe otro desplazamiento τ' tal que $a\mathbf{s} + b\mathbf{s}^\tau = \mathbf{s}^{\tau'}$. Como la sucesión es periódica, se puede suponer $0 \leq \tau' < T$.*

Demostración:
Buscamos probar que el conjunto W de todos los desplazamientos de $\mathbf{s}$, junto con la sucesión nula, es un subespacio vectorial en el $\mathbb{Z}$-espacio vectorial V de las sucesiones con periodo T. Ya sabemos que este conjunto es cerrado bajo la suma, falta ver que $-\mathbf{s} = \mathbf{s}^{\tau'}$ para cierto desplazamiento τ'.
Basta observar que, por la propiedad de desplazamiento y suma, sumar $\mathbf{s}$ a un elemento de W es una biyección de W en si mismo. En particular, existe algún desplazamiento τ' tal que $\mathbf{s} + \mathbf{s}^{\tau'}$ es la sucesión nula. □

Queda plantear un pequeño Lema para que podamos probar el último resultado de esta sección.

Lema 3.26. *Sea* $\mathbf{s}$ *una sucesión periódica con la propiedad de desplazamiento y suma. Fijado un carácter no trivial* χ*, la función de autocorrelación de* $\mathbf{s}$ *para cualquier desplazamiento* τ *es igual al desequilibrio de* $\mathbf{s}$ *respecto a dicho carácter:*

$$\mathcal{A}_{\mathbf{s}}(\tau) = Z_{\chi}(\mathbf{s}).$$

Demostración:
Operemos

$$\mathcal{A}_{\mathbf{s}}(\tau) = \sum_{i=0}^{T-1} \chi(s_i)\overline{\chi(s_{i+\tau})} = \sum_{i=0}^{T-1} \chi(s_i - s_{i+\tau}) = \sum_{i=\tau'}^{T-1+\tau'} \chi(s_i) = Z_{\chi}(\mathbf{s}),$$

donde en la penúltima igualdad se ha utilizado la Proposición 3.25. □

Proposición 3.27. *Sea* $\mathbf{s}$ *una sucesión periódica. Si se verifican:*

1. $\mathbf{s}$ *está equilibrada,*
2. $T \equiv a \bmod |\mathcal{W}|$*, con* $a = -1,\ 0$ *o* 1*,*
3. $\mathbf{s}$ *tiene la propiedad de desplazamiento y suma,*

entonces $\mathbf{s}$ *tiene función de autocorrelación ideal.*

Demostración:
Es inmediato a partir del Lema anterior y de la Proposición 3.14. □

En conclusión: ser de de Bruijn, junto con la propiedad de desplazamiento y suma, constituirán dos herramientas fundamentales para probar el resto de propiedades que hemos introducido. Con todo esto, ya está dispuesto el escenario para los llamados "LFSRs" en el siquiente Capítulo.

Capítulo 4

Registros de Desplazamiento con Retroalimentación Lineal (LFSRs)

0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, ...

LFSR con polinomio $x^4 + x + 1$

En este Capítulo introduciremos los llamados "LFSRs". Estos dispositivos generan sucesiones con buenas propiedades estadísticas, lo cuál los hace extremadamente valiosos para la criptografía en flujo. Los libros de referencia son [25][Capítulo 6] y [16][Capítulo 3], aunque en general no se deben esperar demasiadas similitudes.

4.1. Introducción

Los fenómenos naturales exhiben frecuentemente un comportamiento caótico, que se considera "verdaderamente aleatorio" para todo propósito razonable. Este caos, es decir, la imposibilidad de predecir la trayectoria del sistema a partir de las condiciones iniciales, proviene de la ecuación diferencial que gobierna la evolución del sistema. Con ánimo de imitar a la naturaleza, es por tanto razonable considerar sucesiones numéricas que se "retroalimentan" como candidatas para generar una sucesión de números pseudoaleatoria.
En ese sentido, las leyes de recurrencia lineales son las más sencillas que hay. Tanto la linealidad como la iteración son ideas centrales en Matemáticas, así que no es de extrañar que las sucesiones que llamaremos "recursivas lineales" aparezcan en áreas muy diversas, más allá de su uso en Criptografía. Por ejemplo, la conocida sucesión de Fibonacci tiene una ley de recurrencia lineal: se construye fijando los dos primeros términos $a_0 = 1$, $a_1 = 1$, y formando los siguientes de manera recursiva, $a_n = a_{n-1} + a_{n-2}$ para $n \geq 2$. Esta sucesión, junto con otras análogas (sucesión de Lucas, sucesión de Pell, sucesión de "Tribonacci"...), surgen a menudo como solución a ciertos problemas en Combinatoria. Tal es su importancia que se ha establecido una revista científica (The Fibonacci Quarterly, https://www.fq.math.ca/) dedicada exclusivamente a publicar artículos relacionados con ella.
Por otro lado, en el área de Sistemas Dinámicos, las sucesiones recursivas lineales aparecen como soluciones de las llamadas ecuaciones en diferencias lineales con coeficientes constantes. Estas "ecuaciones en diferencias" son, en cierto modo, los análogos discreto a las ecuaciones diferenciales. De hecho, en Análisis Numérico es usual aproximar la solución de una ecuación diferencial considerando una ecuación en diferencias asociada (métodos lineales multipaso, aunque aquí los coeficientes ya no son constantes).
Todos los casos anteriores son sucesiones sobre $\mathbb{Z}$ o $\mathbb{R}$. Para nuestros propósitos, es más adecuado operar en un cuerpo finito $\mathbb{F}_q$. Si bien es posible trabajar con más generalidad, en las

aplicaciones prácticas rara vez trabajaremos fuera de un cuerpo finito.

Definición 4.1. Sea $\mathbf{s}$ una sucesión de elementos de $\mathbb{F}_q$. Si existe un polinomio no nulo $f \in \mathbb{F}_q[x]$,

$$f(x) = a_k x^k + a_{k-1}x^{k-1} + \ldots + a_0 \not\equiv 0,$$

tal que para todo $n \geq 0$ se verifica

$$a_k s_{n+k} + a_{k-1}s_{n+k-1} + \ldots + a_0 s_n = 0, \tag{4.1}$$

entonces se dice que $\mathbf{s}$ es una sucesión recursiva lineal, y que f genera a $\mathbf{s}$. Por convenio, el polinomio idénticamente nulo $0 \in \mathbb{F}_q[x]$ genera cualquier sucesión.

Nota 4.2. Como $a_k \neq 0$, podemos considerar el polinomio $\tilde{f} = f/a_k$,

$$\tilde{f}(x) = x^k + (a_{k-1}/a_k)x^{k-1} + \ldots + (a_0/a_k),$$

que también genera $\mathbf{s}$. Notemos además que si $a_{j-1} = \ldots = a_0 = 0$, los j primeros términos de $\mathbf{s}$ pueden ser arbitrarios, y entonces el polinomio $\tilde{\tilde{f}} = \tilde{f}/x^j$,

$$\tilde{\tilde{f}}(x) = x^{k-j} + (a_{k-1}/a_k)x^{k-1-j} + \ldots + (a_j/a_k),$$

genera la sucesión desplazada $\mathbf{s}^j$. En conclusión, no hay pérdida de generalidad si suponemos $a_k = 1$, y tampoco cambian las sucesiones, salvo un número finito de términos, si consideramos $a_0 \neq 0$. Este será nuestro convenio de aquí en adelante, si bien no coincide con el de [25]).
De este modo, si $f \in \mathbb{F}_q[x]$ genera una sucesión $\mathbf{s}$, tenemos para todo $n \geq 0$ la siguiente ley de recurrencia:

$$s_{n+k} = -(a_{k-1}/a_k)s_{n+k-1} - (a_{k-2}/a_k)s_{n+k-2} - \ldots - (a_0/a_k)s_n.$$

Aquí, los elementos $s_0, s_1, \ldots, s_{k-1}$ son las condiciones iniciales de la ley de recurrencia, el resto de elementos quedan determinados a partir de ellos.

Observación 4.3. Las sucesiones recursivas lineales que hemos definido suelen venir acompañadas por el adjetivo "homogéneas". También se pueden definir las sucesiones recursivas lineales no homogéneas, es decir, aquellas sucesiones $\mathbf{s}$ de elementos de $\mathbb{F}_q$ para las que existe un polinomio no nulo $f \in \mathbb{F}_q[x]$,

$$f(x) = a_k x^k + a_{k-1}x^{k-1} + \ldots + a_0 \not\equiv 0,$$

y un elemento del cuerpo, $a \in \mathbb{F}_q$, tal que para todo $n \geq 0$ se verifica

$$a_k s_{n+k} + a_{k-1}s_{n+k-1} + \ldots + a_0 s_n + a = 0.$$

Estas sucesiones parecen más generales, pero en realidad, se reducen al caso homogéneo. Veamos, para todo $n \geq 0$ tenemos

$$\begin{aligned} a_k s_{n+k} + a_{k-1}s_{n+k-1} + a_{k-2}s_{n+k-2} + \ldots + a_0 s_n + a = 0, \\ a_k s_{n+k+1} + a_{k-1}s_{n+k} + a_{k-2}s_{n+k-1} + \ldots + a_0 s_{n+1} + a = 0, \end{aligned}$$

restando la primera igualdad a la segunda

$$a_k s_{n+k+1} + (a_{k-1} - a_k)s_{n+k} + (a_{k-2} - a_{k-1})s_{n+k-1} + \ldots + (a_0 - a_1)s_{n+1} - a_0 s_n = 0,$$

por lo que la sucesión propuesta es en realidad una sucesión recursiva lineal homogénea generada por un polinomio de grado $k+1$. Por consiguiente, no hay pérdida de generalidad en tratar solo el caso $a = 0$.

Ya nos es posible enunciar algún resultado fácil sobre las sucesiones recursivas lineales:

Proposición 4.4. *Sean* $\mathbf{s}^1$, $\mathbf{s}^2$ *dos sucesiones en* $\mathbb{F}_q$ *generadas por un polinomio* $f \in \mathbb{F}_q[x]$. *Para cualquier par de coeficientes* $\alpha_1, \alpha_2 \in \mathbb{F}_q$, *y para cualquier par de desplazamientos* $\tau_1, \tau_2 \in \mathbb{Z}$, *la sucesión* $\alpha_1(\mathbf{s}^1)^{\tau_1} + \alpha_2(\mathbf{s}^2)^{\tau_2}$ *también es generada por* f.

Demostración:
Es inmediato, ya que si f genera $\mathbf{s}_1$ y $\mathbf{s}_2$, también genera cualquier desplazamiento de estas, y por otra parte, la condición 4.1 es lineal sobre la sucesión. □

Proposición 4.5. *Sea* $\mathbf{s}$ *una sucesión recursiva lineal en* $\mathbb{F}_q$. *El conjunto de polinomios que generan* $\mathbf{s}$ *forma un ideal principal* $I = (m)$ *en* $\mathbb{F}_q[x]$.
Si tomamos m *mónico, se dice que* m *es el polinomio mínimo de la sucesión* $\mathbf{s}$. *También se define el orden de* $\mathbf{s}$ *como el grado de su polinomio mínimo* m.

Demostración:
Una sucesión $\mathbf{s}$ es generada por $f \in \mathbb{F}_q[x]$ si y solo si, para todo $n \geq 0$, el homomorfismo (viendo $\mathbb{F}_q[x]$ como un $\mathbb{F}_q$-espacio vectorial)

$$\Phi_n \colon \mathbb{F}_q[x] \to \mathbb{F}_q$$
$$a_k x^k + \ldots + a_1 x + a_0 \mapsto a_k s_{k+n} + \ldots + a_1 s_{1+n} + a_0 s_n$$

se anula en f.
Primero notemos que todos estos homomorfismos verifican que $\Phi_{n+1}(f) = \Phi_n(xf)$ para cualquier $f \in \mathbb{F}_q[x]$. En particular, si f genera $\mathbf{s}$, cualquier $x^m f$ también genera $\mathbf{s}$. Por otro lado, si f_1 y f_2 son dos polinomios que generan $\mathbf{s}$, y $\alpha_1, \alpha_2 \in \mathbb{F}_q$, entonces $\Phi_n(\alpha_1 f_1 + \alpha_2 f_2) = \alpha_1 \Phi_n(f_1) + \alpha_2 \Phi_n(f_2) = 0$ para todo $n \geq 0$, lo que significa que $\alpha_1 f_1 + \alpha_2 f_2$ sigue generando $\mathbf{s}$. Con estos dos hechos se deduce que I es un ideal.
Para terminar, recordemos que $\mathbb{F}_q$ es un cuerpo. Esto implica que $\mathbb{F}_q[x]$ es un dominio de ideales principales, y en consecuencia, $I = (m)$ para cierto $m \in \mathbb{F}_q[x]$, que podemos tomar mónico. □

4.2. LFSRs y complejidad lineal

A continuación explicaremos la procedencia del nombre de este Capítulo.

Observación 4.6. Ya describimos porqué las sucesiones recursivas lineales eran "recursivas". El siguiente diagrama ilustra cómo un polinomio $f \in \mathbb{F}_q[x]$ genera una sucesión recursiva lineal, desplazándose hacia la derecha a cada paso.

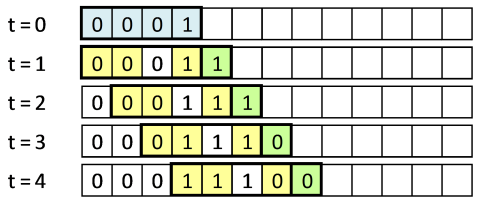

Figura 4.1: El polinomio $f = x^4 + x^3 + x + 1$, donde $f \in \mathbb{F}_2$, genera una sucesión recursiva lineal $\mathbf{s}$ a partir de las condiciones iniciales $s_0 = 0$, $s_1 = 0$, $s_2 = 0$ y $s_3 = 1$ (en azul). Los elementos en amarillo se suman en cada iteración para obtener el siguiente elemento, que marcaremos en verde. Las unidades de tiempo son arbitrarias.

Gran parte del interés en las sucesiones recursivas lineales radica en que se pueden implementar de manera extremadamente eficiente, sobre todo a nivel de hardware. Si el polinomio f tiene grado k, basta tener k registros que almacenen un valor en $\mathbb{F}_q$, a lo sumo k multiplicadores por una constante y $k-1$ sumadores en dicho cuerpo, y un reloj que sincronice el circuito. De esta forma, si tenemos una sucesión recursiva lineal $\mathbf{s}$ generada por f, basta introducir los términos $s_0, \ldots, s_{k-1}$ como condiciones iniciales para que el circuito genere el resto de la sucesión, a razón de un término por pulso de reloj.
Esta configuración se denomina Registro de Desplazamiento con Retroalimentación Lineal. En este trabajo preferiremos el término inglés, Linear Feedback Shift Register, que abreviaremos por "LFSR".
Puesto que todos los circuitos electrónicos operan en binario, los LFSRs sobre cuerpos de característica 2 son los más empleados. Un caso especialmente sencillo ocurre cuando trabajamos en $\mathbb{F}_2$ (como en la Fig. 4.2): entonces se puede prescindir del producto por una constante (o bien entra el término en la suma, o no), y la suma se reduce a la clásica operación XOR entre dos bits.

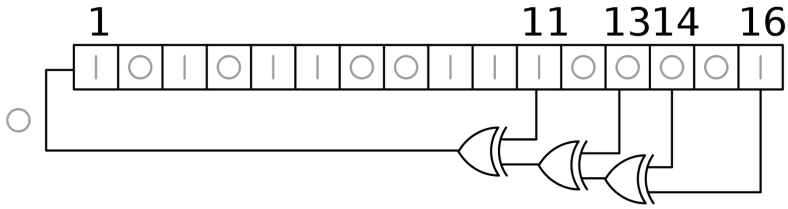

Figura 4.2: Esquema del circuito de un LFSR sobre $\mathbb{F}_2$ asociado al polinomio $x^{15} + x^{13} + x^{12} + x^{10} + 1$. Imagen tomada de [1].

La figura anterior ilustra el modo "estándar" o "de Fibonacci" para implementar un LFSR en hardware. En algunas casos es preferible, en vez de generar un nuevo término de la sucesión a partir de los anteriores, aplicar cada término recién generado a todos los siguientes términos que vaya a afectar, de manera que poco a poco se vayan sumando sobre el registro correspondiente los sumandos de la relación de recurrencia. Esta es la forma "de Galois" de un LFSR, que se muestra en la siguiente figura.

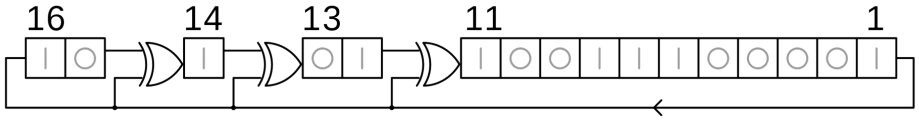

Figura 4.3: Mismo LFSR que en la figura anterior, pero ahora en forma de Galois. En este caso el estado interno debe interpretarse con más cuidado, solo los registros a la derecha del todo han recibido ya todos sus sumandos y son términos de la sucesión. Imagen tomada de [1].

En ocasiones, en vez de considerar una sucesión particular, querremos enfocarnos en un polinomio que genera diversas sucesiones recursivas lineales. A raíz de esta Observación, abusaremos de la notación y llamaremos "LFSR" al polinomio en dichos casos. Si una sucesión es generada

por un polinomio f de grado k, diremos que $\mathbf{s}$ es generada por un LFSR f de longitud k con vector de condiciones iniciales $(s_0, s_1, \ldots, s_{k-1})$.

Nota 4.7. Los polinomios empleados en la operación práctica nunca alcanzan las cotas planteadas en la Observación anterior. Por lo contrario, se procura utilizar LFSRs trinomiales (es decir, cuyas leyes de recurrencia con a lo sumo tres a_i no nulos), que son los LFSRs no triviales más rápidos y menos costosos en ese sentido.

Es razonable plantearse qué ocurriría si invirtiéramos el sentido de un LFSR. Esto viene cubierto por el concepto de "polinomio recíproco".

Definición 4.8. Dado un polinomio $f \in \mathbb{F}_q[x]$ de grado k, se define el polinomio recíproco de f como $f^\perp = x^k g(x^{-1})$, donde se ve f en el anillo de polinomios de Laurent $\mathbb{F}_q(x)$. Si f tiene la forma

$$f = a_k x^k + a_{k-1} x^{k-1} + \ldots + a_0,$$

entonces $f^\perp$ tiene la forma

$$f^\perp = a_0 x^k + a_1 x^{k-1} + \ldots + a_k. \tag{4.2}$$

Observación 4.9. Como se deduce inmediatamente de 4.2, si se tiene una sucesión recursiva lineal generada por un polinomio f, la sucesión recursiva lineal generada por el polinomio $f^\perp$ es la misma, pero con el orden "invertido".
Más precisamente, dada una sucesión recursiva lineal $\mathbf{s}$ generada por f, y otra sucesión recursiva lineal $\mathbf{s}'$ generada por $f^\perp$, si para ciertos n, m naturales con $n \geq 1$ se verifica

$$s'_m = s_{n+k-1}, s'_{m+1} = s_{n+k-2} \ldots, s'_{m+k-1} = s_n,$$

entonces el siguiente término de $\mathbf{s}'$ será el anterior término de $\mathbf{s}$. Es decir, $s'_{m+k} = s_{n-1}$, y sucesivamente $s'_{m+k+1} = s_{n-2}$, $s'_{m+k+2} = s_{n-3}$, ..., hasta que lleguemos a s_0 (de hecho, no habría ningún problema en definir las sucesiones recursivas lineales con índices en todo $\mathbb{Z}$, pero esto estaría menos alineado con la operación de los LFSRs en circuitos/programas reales).

Ejemplo 4.10. Ilustramos lo anterior.

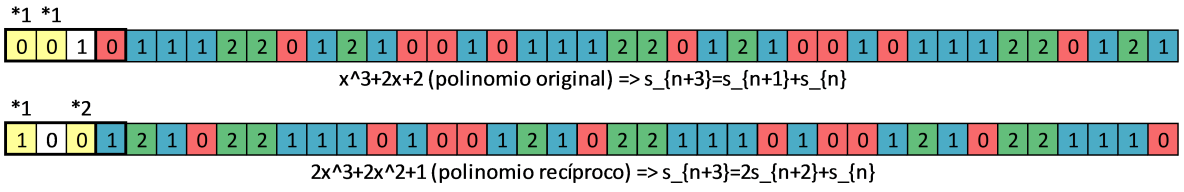

Figura 4.4: Un LFSR sobre $\mathbb{F}_3$ asociado al polinomio $x^3 + 2x + 2$, y el LFSR recíproco, asociado al polinomio $2x^3 + 2x^2 + 1$. Ahora se marca la relación de recurrencia en las condiciones iniciales, el resto de la sucesión se da directamente (con distintos colores cada número, para mejor visualización).

Las dos sucesiones son periódicas (más adelante veremos que toda sucesión recursiva lineal es periódica), y efectivamente la sucesión de abajo es la de arriba en orden inverso. Esto ocurre porque hemos puesto como condición inicial del segundo LFSR la condición inicial invertida del primero (en realidad no era necesario en este caso, como veremos posteriormente).

Ahora daremos una definición importante motivada por la Observación 4.6.

Definición 4.11. Dada una sucesión periódica $\mathbf{s}$ en $\mathbb{F}_q$, se define su complejidad lineal $LC(\mathbf{s})$ como el grado de su polinomio mínimo.
La complejidad lineal está bien definida, pues toda sucesión periódica es una sucesión recursiva lineal: si T es el periodo de $\mathbf{s}$, entonces $\mathbf{s}$ es generada por el polinomio $x^T - 1$.

Finalmente, proporcionaremos un par de cotas básicas para la complejidad lineal. Cambiamos ligeramente la notación para dos sucesiones distintas, con el objetivo de evitar confusiones cuando se involucren potencias.

Proposición 4.12. *La complejidad lineal de la suma de dos sucesiones periódicas es menor o igual que la suma de sus complejidades lineales.*

Demostración:
Sean $\mathbf{s}^{(1)}$ y $\mathbf{s}^{(2)}$ las dos sucesiones periódicas. Como son periódicas, son recursivas lineales, sean f_1 y f_2 sus polinomios mínimos. Si definimos $f = f_1 f_2$, entonces $f_1 \mid f$, por lo que f genera $\mathbf{s}^{(1)}$, y $f_2 \mid f$, por lo que f genera $\mathbf{s}^{(2)}$. En consecuencia, por la Proposición 4.4, f genera $\mathbf{s}^{(1)} + \mathbf{s}^{(2)}$. □

Proposición 4.13. *La complejidad lineal del producto de dos sucesiones periódicas es menor o igual que el producto de sus complejidades lineales*

Demostración:
Esta prueba está basada en [23][Teorema 3.5]. Ahí se puede encontrar bien hecha, en el caso probable de que yo haya cometido algún error aquí por las prisas. Alternativamente, es más agradable, pero también más larga, la prueba en [39].
Sean $\mathbf{s}^{(1)}$ y $\mathbf{s}^{(2)}$ las dos sucesiones periódicas, $f^{(1)}$ y $f^{(2)}$ sus polinomios mínimos, sean L_1 y L_2 sus complejidades lineales, y sean

$$s_n^{(1)} = \sum_{i=0}^{N_1^{(1)}} b_{i,1}^{(1)} (\gamma_i^{(1)})^n + \sum_{i=0}^{N_2^{(1)}} b_{i,2}^{(1)} n (\gamma_i^{(1)})^n + \ldots + \sum_{i=0}^{N_k^{(1)}} b_{i,k}^{(1)} n^{k-1} (\gamma_i^{(1)})^n,$$
$$s_n^{(2)} = \sum_{i=0}^{N_1^{(2)}} b_{i,1}^{(2)} (\gamma_i^{(2)})^n + \sum_{i=0}^{N_2^{(2)}} b_{i,2}^{(2)} n (\gamma_i^{(2)})^n + \ldots + \sum_{i=0}^{N_k^{(2)}} b_{i,k}^{(2)} n^{k-1} (\gamma_i^{(2)})^n,$$

las fórmulas generales 4.4 de los términos de cada sucesión (estamos adelantando ese resultado, pero no hay problema, ya que es independiente de este otro). Si definimos $e_j^{(l)} = \max\{i \mid N_i^{(l)} \geq j\}$, entonces la fórmula general de los términos de la sucesión producto $\mathbf{s}^{(1)}\mathbf{s}^{(2)}$ es

$$s_n^{(1)} s_n^{(2)} = \sum_{i_1=0}^{N_1^{(1)}} \sum_{i_2=0}^{N_1^{(2)}} p_{i_1,i_2}(n) (\gamma_{i_1}^{(1)} \gamma_{i_2}^{(2)})^n, \tag{4.3}$$

donde cada p_{i_1,i_2} es un polinomio en $\mathbb{F}_q[x]$ de grado $(e_i^{(1)} - 1) + (e_i^{(2)} - 1)$.
Finalmente, afirmamos que el polinomio

$$f(x) = \prod_{i_1=0}^{N_1^{(1)}} \prod_{i_2=0}^{N_1^{(2)}} (x - \gamma_{i_1}^{(1)} \gamma_{i_2}^{(2)})^{e_i^{(1)} + e_i^{(2)} - 1}$$

genera cada sumando de 4.3 (esto es rutina, si bien algo tedioso; se hace por inducción sobre el exponente que acompaña a $(x - \gamma_{i_1}^{(1)}\gamma_{i_2}^{(2)})$), y, en consecuencia, genera $\mathbf{s}^{(1)}\mathbf{s}^{(2)}$. Este polinomio tiene grado

$$\sum_{i_1=0}^{N_1^{(1)}}\sum_{i_2=0}^{N_1^{(2)}}(e_{i_1}^{(1)} + e_{i_2}^{(2)} - 1) \leq \sum_{i_1=0}^{N_1^{(1)}} e_i^{(1)} \sum_{i_2=0}^{N_1^{(2)}} e_{i_2}^{(2)} \leq L_1 L_2,$$

y está en $\mathbb{F}_q[x]$ por simetría (al aplicar el endomorfismo de Frobenius, solo reorganizamos las raíces de los distintos factores irreducibles de $f^{(1)}$ y $f^{(2)}$) (incluso si no lo estuviera, no habría problema, bastaría realizar la proyección de $\mathbb{F}_{q^{n!}}$ en su subespacio lineal $\mathbb{F}_q$). □

4.3. Forma matricial

Muchas propiedades de periodicidad de las sucesiones recursivas lineales se entienden mejor cuando estas se escriben en forma matricial, lo que constituye el enfoque algebraico de los LFSRs. Cabe mencionar que también son posibles otros puntos de vista, como el de las funciones generatrices o el de la traza, que nosotros no consideraremos en este Trabajo de Fin de Grado.

Definición 4.14. Sea $\mathbf{s}$ una sucesión recursiva lineal generada por un polinomio $f \in \mathbb{F}_q[x]$,

$$f(x) = x^k + a_{k-1}x^{k-1} + \ldots + a_1 x + a_0,$$

que consideraremos fijo. Se define la sucesión vectorial $\mathbf{S}$ asociada a $\mathbf{s}$ como

$$S_n = (s_n, s_{n+1}, \ldots, s_{n+k-2}, s_{n+k-1})^t,$$

para todo $n \geq 0$. Esta sucesión obedece la ley de recurrencia $S_{n+1} = AS_n$ para todo $n \geq 0$, donde A es la matriz compañera del polinomio f,

$$A = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -a_0 & -a_1 & -a_2 & \cdots & -a_{k-1} \end{bmatrix}$$

Como consecuencia de esta relación de recurrencia, $S_i = A^i S_0$ para todo $i \geq 0$.

La Proposición 4.16, entre otras cosas, justifica que en algunos textos se llame "polinomio característico" a cualquier polinomio que genere una sucesión recursiva lineal.

Nota 4.15. Durante todo el texto, se toma el convenio de que $|xI - A|$ es el polinomio característico de A (I es la matriz identidad de dimensión k).

Proposición 4.16. *Sea K un cuerpo, sea A la matriz compañera de un polinomio $f \in K[x]$ con $x \nmid f$ y $\deg(f) = k > 1$. Se verifican las siguiente afirmaciones:*

1. *f es el polinomio característico y el polinomio mínimo de A.*
2. *A es invertible, y si a_0 es el término independiente de f, entonces $((-1)^k/a_0)f^{\perp}$ es el polinomio característico y el polinomio mínimo de A^{-1}. Esto último es cierto incluso si solo se exige que f sea el polinomio característico de A.*

Demostración: 1. Comprobar que f es el polinomio característico de A es trivial con la fórmula combinatoria para los determinantes. Si se decide utilizar la fórmula recurrente para los determinantes, basta entonces una simple inducción sobre k.
No hace falta utilizar el Teorema de Cayley-Hamilton para ver que f anula a A. Basta recordar que $AS_n = S_{n+1}$ para todo $n \geq 0$, y como f genera $\mathbf{s}$, todas las entradas de $f(A)S_0$ se deben anular. Pero S_0 es un vector arbitrario, así que $f(A) = 0$.
Falta ver que el polinomio mínimo de A debe tener grado k. Veamos, si consideramos la condición inicial $S_0 = (0, \dots, 0, 1)^t$, los primeros términos k términos de la sucesión vectorial tienen la forma $S_i = (0, \dots, 0, \overset{i)}{1}, s_k, \dots, s_{k+i})^t$, por lo que son linealmente independientes. Entonces ningún polinomio g de grado $k-1$ puede anular A, ya que $g(A)S_0$ es una combinación lineal con coeficientes no todos nulos de los k vectores S_i.

2. Como $a_0 \neq 0$, todas las filas de A son linealmente independientes, por lo que A es invertible. En particular, $|A| = a_0$. Con esto, el polinomio característico de la matriz inversa A^{-1} es:

$$\begin{aligned} |xI - A^{-1}| &= x^k |I - x^{-1}A^{-1}| \\ &= x^k |(A - x^{-1}I)A^{-1}| \\ &= x^k |A - x^{-1}I||A^{-1}| = x^k (-1)^k f(x^{-1})|A|^{-1}. \end{aligned}$$

Para probar que también es el polinomio mínimo, basta notar que, como A es invertible, para todo $g \in \mathbb{F}_q[x]$ son equivalentes $g(A) = 0$ y $g^{\perp}(A^{-1}) = (A^{-1})^k g(A) = 0$.
Si solo nos interesa el caso particular de A matriz compañera de un polinomio con $a_0 \neq 0$, cabe señalar otra vía: podemos calcular explícitamente la inversa de A sin mucha dificultad, queda una matriz con forma parecida a A, y aplicando argumentos análogos a los anteriores se halla su polinomio característico y se prueba que también es el mínimo.

□

Terminamos con una observación interesante, que abre una vía alternativa a la que usaremos en la siguiente sección:

Observación 4.17. Sea $\mathbf{s}$ una sucesión recursiva lineal en $\mathbb{F}_q$. En el caso particular de que el polinomio mínimo m de la sucesión sea irreducible (que será el caso más interesante), todas sus raíces son distintas, y por tanto, A diagonaliza en $\mathbb{F}_{q^k}$. A partir de dicha diagonalización, junto con la Proposición A.27, se obtiene una fórmula para el término general de la sucesión,

$$s_n = \sum_{i=0}^{k-1} b_i \gamma^{in}$$

para ciertos $b_0, \dots, b_{k-1} \in \mathbb{F}_q$, y para cierto $\gamma \in \mathbb{F}_{q^k}$.
Con más generalidad, podemos considerar la forma de Jordan de A en $\mathbb{F}_{q^{k!}}$ (un cuerpo lo suficientemente grande para contener todas las ráices de cualquier polinomio de grado k o menor). Recordemos que esta es una matriz a A con la forma

$$A = \begin{bmatrix} J_1 & & & \\ & J_2 & & \\ & & \ddots & \\ & & & J_p \end{bmatrix}, \qquad J_i = \begin{bmatrix} \lambda_i & 1 & & \\ & \lambda_i & \ddots & \\ & & \ddots & 1 \\ & & & \lambda_i \end{bmatrix}.$$

En esta notación se pueden repetir los λ_i. En todo caso, es bien conocida la fórmula para la potencia de un bloque de Jordan (m_i aquí denota el tamaño del bloque):

$$(J_i)^n = \begin{bmatrix} \lambda_i^n & \binom{n}{1}\lambda_i^{n-1} & \binom{n}{2}\lambda_i^{n-2} & \cdots & \cdots & \binom{n}{m_i-1}\lambda_i^{n-m_i+1} \\ & \lambda_i^n & \binom{n}{1}\lambda_i^{n-1} & \cdots & \cdots & \binom{n}{m_i-2}\lambda_i^{n-m_i+2} \\ & & \ddots & \ddots & \vdots & \vdots \\ & & & \ddots & \ddots & \vdots \\ & & & & \lambda_i^n & \binom{n}{1}\lambda_i^{n-1} \\ & & & & & \lambda_i^n \end{bmatrix}.$$

A partir de esta relación, y teniendo en cuenta que todos los λ_i son invertibles (pues $a_0 \neq 0$), se obtiene una fórmula para el término general de la sucesión,

$$s_n = \sum_{i=0}^{N_1} b_{i,1}\gamma_i^n + \sum_{i=0}^{N_2} b_{i,2} n\gamma_i^n + \ldots + \sum_{i=0}^{N_k} b_{i,k} n^{k-1}\gamma_i^n, \tag{4.4}$$

donde las γ_i son raíces en $\mathbb{F}_{q^{n!}}$ distintas, los N_j son el número de raíces con bloques de tamaño j o mayor, y los $b_{i,j}$ son coeficientes en $\mathbb{F}_q$.
Como nota aparte, la forma de Jordan también explica de manera natural porqué un mismo LFSR puede dar lugar a sucesiones recursivas lineales con distinto periodos, ya que las condiciones iniciales se pueden distribuir de maneras muy distintas entre los bloques.

4.4. Periodo de un LFSR

Una de las características esenciales que se debe pedir a cualquier candidato de sucesión pseudoaleatoria es que tenga un periodo muy grande. Veamos qué se puede decir sobre el periodo de las sucesiones recursivas lineales y de los LFSRs.

Definición 4.18. Definimos el periodo máximo de un LFSR como el máximo periodo de una sucesión recursiva lineal generada por este.

Nota 4.19. Aquí seguimos manejando la Definición 3.2. En [25] se hace la distinción entre los varios "periodos" de la sucesión, y el "periodo mínimo" de esta, y se introducen otros conceptos como "eventualmente periódico" y "preperiodo". Nosotros hemos simplificado la terminología y no tratamos el caso no homogéneo, todo para evitar sobrecargar la notación innecesariamente.

Observación 4.20. Las sucesiones generadas por un mismo LFSR pueden tener distintos periodos dependiendo de las condiciones iniciales. Por ejemplo, si consideramos el LFSR en 4.1, dependiendo de las condiciones iniciales, se obtienen sucesiones recursivas lineales de periodo 2, 3 y 6 (además de la sucesión trivial con todo ceros, que tiene periodo 1).

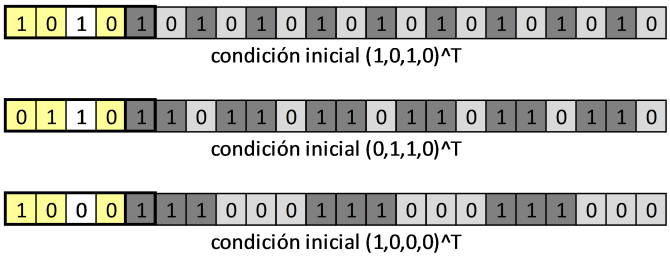

Figura 4.5: Sucesiones recursivas lineales correspondientes a las condiciones iniciales $(1,0,1,0)^t$, $(0,1,1,0)^t$, y $(1,0,0,0)^t$ respectivamente. El LFSR es el asociado al polinomio $f = x^4 + x^3 + x + 1$, donde $f \in \mathbb{F}_2[x]$. De aquí en adelante, todos los Ejemplos los consideraremos sobre $\mathbb{F}_2$ por simplicidad.

En la Definición 4.11 mencionamos que, trivialmente, toda sucesión periódica es una sucesión recursiva lineal. Veamos que el recíproco también es cierto.

Proposición 4.21. *Toda sucesión recursiva lineal es periódica. Si la sucesión es generada por un polinomio de grado k, el periodo es a lo sumo $q^k - 1$.*

Demostración:
Primero observemos que la sucesión recursiva lineal $\mathbf{s}$ es periódica si y solo si lo es la sucesión vectorial asociada $\mathbf{S}$, y en este caso, ambas tienen el mismo periodo. Supongamos que el estado inicial S_0 de la sucesión recursiva vectorial asociada no es el vector nulo $(0, \dots, 0)^t$. Entonces todos los estados, y en particular los q^k primeros $S_0, AS_0, \dots, A^{q^k-1}S_0$, están en $\mathbb{F}_q^k \setminus (0, \dots, 0)^t$. Este conjunto tiene $q^k - 1$ elementos, por lo que debe haber algún $0 \leq r_1 < r_2 \leq q^k - 1$ tal que $S_{r_1} = S_{r_2}$, y por ser A invertible, $S_{n+(r_2-r_1)} = S_n$ para todo $n \geq 0$. En particular, $r_2 - r_1 \leq q^k - 1$. □

Nota 4.22. Si admitiéramos las sucesiones recursivas lineales no homogéneas de la Observación 4.3, el argumento anterior se adapta permitiendo el vector nulo, y el periodo mínimo está acotado superiormente por q^k. Como enseguida veremos que las sucesiones recursivas lineales "homogéneas" alcanzan la cota de la Proposición se alcanza, concluimos que no hay ninguna beneficio real en considerar las "no homogéneas".

Proposición 4.23. *El periodo de una sucesión generada por un LFSR divide al orden de la matriz asociada A (el orden multiplicativo, como elemento del grupo lineal general $GL_k(\mathbb{F}_q)$).*

Demostración:
Sea T el periodo de la sucesión, sea r el orden de A. Entonces $S_{n+r} = A^r S_n = IS_n = S_n$ para todo $n \geq 0$, por tanto $T \mid r$. □

Proposición 4.24. *El periodo máximo de un LFSR se alcanza considerando la sucesión $\mathbf{s}$ generada con condiciones iniciales $S_0 = (0, \dots, 0, 1)^t$. Está sucesión se llama sucesión de respuesta a impulso del LFSR.*

Demostración:
Sea T es el periodo de $\mathbf{S}$, se verifica

$$IS_i = S_i = S_{i+T} = A^T S_i \tag{4.5}$$

para todo $i \geq 0$. Ya sabemos que los S_i con $i = 0, \dots, k-1$ son linealmente independientes. En consecuencia, 4.5 se verifica si y solo si $I = A^T$, de lo que se deduce que $r \mid T$, donde r es el orden de A. Recordemos que $T \mid r$ por la Proposición anterior, así que hemos acabado. □

Ahora enunciamos uno de los resultados importantes de este Capítulo:

Teorema 4.25. *El periodo de una sucesión recursiva lineal es igual al orden de su polinomio mínimo.*

Demostración:
Basta recordar la Proposición 4.5 y que, para sucesiones recursivas lineales, tener periodo T es equivalente a ser generada por el polinomio $x^T - 1$. □

Corolario 4.26. *El periodo máximo de un LFSR es el orden de su polinomio asociado.*

Demostración:
Sea f el polinomio asociado al LFSR. A la vista del Teorema anterior y de la Proposición 4.24, basta probar que la sucesión recursiva lineal generada por f con condiciones iniciales $S_0 = (0, \dots, 0, 1)^t$ tiene a f como polinomio mínimo. Esto es inmediato, ya que cualquier otro polinomio g con $\deg(g) < k$ tendría como condiciones iniciales el vector nulo, y por tanto, generaría la sucesión nula. □

Ejemplo 4.27. Usando la Proposición 4.24, podemos obtener el periodo máximo de los LFSRs correspondientes a los polinomios en el Ejemplo 1.6, el Ejemplo 1.7, el Ejemplo 1.10 y el Ejemplo 1.12.

Figura 4.6: Periodo máximo de los LFSRs asociados a los polinomios $x^4 + x + 1$, $x^4 + x^3 + x^2 + x + 1$, $x^6 + x^5 + x^3 + x + 1$ y $x^4 + x^3 + x + 1$ respectivamente.

Si bien se puede diseñar un LFSR para cualquier polinomio en $\mathbb{F}_q[x]$, normalmente solo se utilizan los basados en polinomios primitivos. Veamos cuál es el motivo detrás de esta elección.

Definición 4.28. Una sucesión recursiva lineal con polinomio mínimo primitivo se denomina m-sucesión.

Observación 4.29. Ya sabemos que, en el fondo, ser periódica y ser recursiva lineal son condiciones equivalentes para sucesiones. Por tanto, a raíz de la Proposición 4.21 y del Teorema 1.16, las m-sucesiones son las sucesiones de periodo maximal entre las sucesiones recursivas lineales de orden k, y también son las de complejidad lineal minimal entre las sucesiones periódicas de periodo $q^k - 1$.

Las m-sucesiones son una clase reducida y especial de sucesiones. En la siguiente sección precisaremos más sus propiedades, pero quizás por la coincidencia de los periodos el lector se haya percatado ya de que las m-sucesiones son sucesiones de de Bruijn agujereadas. Sabemos por el Teorema 3.23 que realmente existen muchas sucesiones agujereadas de de Bruijn, veremos que sin embargo existen muchas menos m-sucesiones.

Proposición 4.30. *Existen $\varphi(q^k - 1)/k$ m-sucesiones de orden k.*

Demostración:
En el grupo cíclico $\mathbb{F}^*_{q^k}$ existen $\varphi(q^k - 1)$ elementos primitivos, que se reparten para constituir las k raíces de cada polinomio primitivo de grado k. □

4.5. Propiedades de las m-sucesiones

Como generadores de sucesiones pseudoaleatorias, los LFSRs son demasiado simples, y en consecuencia, no lo suficientemente seguros para ser utilizados directamente en Criptografía. Por otro lado, es justo esa simplicidad lo que permite realizar un análisis matemático exhaustivo. En particular, nos interesan las m-sucesiones: como veremos, estas sucesiones pseudoaleatorias verifican todas las propiedades planteadas en la Sección 3. Por esta razón, los LFSRs con polinomio primitivo se utilizan como ladrillo básico para la construcción de otros cifrados en flujo más complejos y seguros.

Teorema 4.31. *Sea* $\mathbf{s}$ *una* m*-sucesión de orden* k*. Se verifican*

1. $\mathbf{s}$ *es una sucesión de de Bruijn agujereada, y en consecuencia,* $\mathbf{s}$ *está equidistribuida hasta orden* k *y cumple la propiedad de las rachas.*

2. $\mathbf{s}$ *satisface la condición de desplazamiento y suma, y en consecuencia,* $\mathbf{s}$ *tiene función de autocorrelación ideal.*

Demostración:

1. Fijémonos en la forma matricial de la sucesión recursiva lineal. Como las condiciones iniciales son no nulas, el bloque nulo de longitud k no aparece en ningún momento. Por otro lado, como cada bloque de longitud k determina la generación del resto de la sucesión, no se puede repetir un bloque antes de que termine un periodo. El periodo T es q^k-1, así que cada uno de los q^k-1 bloques no nulos debe aparecer exactamente una vez dentro de un periodo.

2. Por la Proposición 4.4, el polinomio primitivo m de grado k que genera $\mathbf{s}$ debe generar también la sucesión $\mathbf{s}+\mathbf{s}^\tau$. Recordemos por otro lado que $\mathbf{s}$ es una sucesión de de Bruijn agujereada de orden k, por lo que m genera un desplazamiento de $\mathbf{s}$ para todas las q^k-1 condiciones iniciales no nulas. En particular, si $\mathbf{s}+\mathbf{s}^\tau$ no es la sucesión nula, entonces $\mathbf{s}+\mathbf{s}^\tau$ tiene condiciones iniciales no nulas y es generada por m.

□

El matemático americano Solomon W. Golomb (1932-2016) es conocido principalmente por su trabajo pionero en la Teoría de la Información y de la Codificación, si bien este también realizó importantes aportaciones a la Combinatoria y a la Teoría de Números a lo largo de su vida. En lo que concierne a sucesiones pseudoaleatorias, Golomb enunció tres condiciones que se debía exigir a una sucesión binaria para considerarla "suficientemente aleatoria".

Definición 4.32. Una sucesión $\mathbf{s}$ cumple los postulados de Golomb si

1. Es equilibrada.

2. Cumple la propiedad de la racha.

3. Tiene función de autocorrelación ideal.

Para más información sobre la importancia de estos postulados en Criptografía, consultar [38]. Como hemos visto, las m-sucesiones cumplen estas tres condiciones. Esto seguramente no es casualidad, ya que Golomb fue una de las figuras de referencia en el desarrollo de la teoría de los LFSRs [15].
En esta línea, Golomb se preguntó si existían otras sucesiones que también las cumplieran los postulados. Esta cuestión se respondió de manera afirmativa (ver [16, Capítulo 12]), pero sigue siendo una conjetura para el caso particular de sucesiones binarias (es decir, $\mathcal{W}=\mathbb{F}_2$).

Conjetura 4.33. *Las únicas sucesiones en $\mathbb{F}_2$ que cumplen los postulados de Golomb son las m-sucesiones.*

En [19] se discuten distintas formas (no todas equivalentes) de esta Conjetura. Hasta donde llega el conocimiento del autor de este Trabajo de Fin de Grado, este problema permanece abierto hoy en día.

4.6. Algortimo de Berlekamp-Massey

Los ordenadores son de naturaleza finita, y en ese sentido, solo manejan un número finito de entradas de cualquier sucesión dada. En particular, los LFSRs operan en tiempo finito, y, en consecuencia, generan finitos términos de la sucesión recursiva lineal correspondiente, normalmente ni siquiera terminando el periodo. Introducimos notación para referirnos a esa situación.

Notación 4.34. Durante el resto de este Capítulo se había denotado el grado de un polinomio por k. Ahora hemos cambiado el convenio, y reservamos k para denotar el número de términos de la sucesión truncada. El motivo principal es seguir lo más fielmente posible la notación original, lo que nos podemos permitir ya que realmente solo tendremos que usar una letra alternativa "j" en la siguiente Definición.

Definición 4.35. Sea $\mathbf{s}$ una sucesión, sea $k \geq 1$ un número natural. La sucesión truncada $\mathbf{s}^{[k]}$ es el vector formado por los k primeros términos $s_0, \ldots, s_{k-1}$ de la sucesión.
Si $\mathbf{s}$ es una sucesión en $\mathbb{F}_q$, diremos que un polinomio no nulo $f \in \mathbb{F}_q[x]$,

$$f(x) = a_j x^j + a_{j-1}x^{j-1} + \ldots + a_0 \not\equiv 0,$$

genera la sucesión truncada $\mathbf{s}^{[k]}$ si para todo n con $0 \leq n \leq (k-1) - j$, se verifica

$$a_j s^{[k]}_{n+j} + a_{j-1} s^{[k]}_{n+j-1} + \ldots + a_0 s^{[k]}_n = 0. \tag{4.6}$$

Por convenio, el polinomio idénticamente nulo $0 \in \mathbb{F}_q[x]$ genera cualquier sucesión truncada.
También definimos la complejidad lineal $LC(\mathbf{s}^{[k]})$ de una sucesión truncada $\mathbf{s}^{[k]}$ como el menor grado de un polinomio que la genera. A un polinomio mónico que cumpla esta condición lo llamaremos *un* polinomio mínimo de $\mathbf{s}^{[k]}$, pues no siempre será único.

Observación 4.36. En ambas definiciones (la Definición 4.11 y la anterior), la complejidad lineal es en el fondo el menor tamaño de un LFSR que puede generar la sucesión o sucesión truncada. En particular, cualquier polinomio $f \in \mathbb{F}_q[x]$ con $\deg(f) \geq k$ genera $\mathbf{s}^{[k]}$, ya que si tenemos un registro de tamaño igual o mayor que el vector $\mathbf{s}^{[k]}$, simplemente podemos incluir dicho vector en las condiciones iniciales del LFSR.

Ahora consideramos el problema de hallar el polinomio minimo de una sucesión recursiva lineal. Como normalmente solo dispondremos de una cantidad finita de términos que posiblemente vaya creciendo, reformulamos el problema como hallar un polinomio mínimo de una sucesión recursiva lineal truncada, e ir "actualizándolo" según se reciben nuevos datos. Este es el problema que resuelve de forma eficiente el algoritmo de Berlekamp-Massey, debido al matemático americano Elwyn Berlekamp, que inventó el algoritmo para códigos correctores, y al matemático americano James Massey, que reconoció su utilidad para los LFSRs.
El objetivo de esta sección es enunciar y probar la corrección del algoritmo de Berlekamp-Massey, tomando como referencia [31][Lección 5]. Para ello, necesitaremos realizar algo de trabajo preliminar.

Observación 4.37. Los argumentos de la Proposición 4.5 son parecidos para las sucesiones truncadas. La diferencia principal es que ahora no siempre es admisible la suma de de dos polinomios, ya que puede disminuir el grado, y con ello, aumentar el número de condiciones que debe satisfacer el polinomio suma. Por otro lado, mientras no disminuya el grado, el resto de argumentos es idénticos. En consecuencia, si $f \in \mathbb{F}_q[x]$ es un polinomio que genera la sucesión truncada $\mathbf{s}^{[j]}$ en $\mathbb{F}_q$, y $g \in \mathbb{F}_q[x]$ es otro polinomio, también se verifica que el producto fg genera $\mathbf{s}^{[j]}$. Lo único que sucede es que el conjunto de polinomios que generan una sucesión truncada ya no es necesariamente un ideal. Otras propiedades de la complejidad lineal sí se conservan para sucesiones truncadas. En particular, se fácil reproducir la prueba de la Proposición 4.12 para comprobar que se verifica un resultado análogo, lo que se utilizará para demostrar el siguiente Lema.

Lema 4.38. *Sea* $\mathbf{s}$ *una sucesión en* $\mathbb{F}_q$*, sea* $\mathbf{s}^{[k]}$ *una truncación de la sucesión, sea* $f^{[k]}$ *un polinomio mínimo de* $\mathbf{s}^{[k]}$*, y supongamos que* $f^{[k]}$ *no genera* $s^{[k+1]}$*. Entonces*

$$LC(\mathbf{s}^{[k+1]}) \geq \text{máx}\{LC(\mathbf{s}^{[k]}), k+1-LC(\mathbf{s}^{[k]})\}.$$

Demostración:
Es evidente que $LC(\mathbf{s}^{[k+1]}) \geq LC(\mathbf{s}^{[k]})$, ya que cualquier polinomio que genere $\mathbf{s}^{[k+1]}$ también genera $\mathbf{s}^{[k]}$. Por otro lado, como $f^{[k]}$ no genera $\mathbf{s}^{[k+1]}$, debe existir $a \in \mathbb{F}_q$, $a \neq 0$ tal que $f^{[k]}$ genera $\mathbf{s}^{[k+1]} + (0, \overset{(k \text{ ceros})}{\ldots}, 0, a)$, es decir,

$$LC(\mathbf{s}^{[k+1]} + (0, \ldots, 0, a)) = LC(\mathbf{s}^{[k]}).$$

En consecuencia,

$$\begin{aligned} LC(\mathbf{s}^{[k+1]}) + LC(\mathbf{s}^{[k]}) &= LC(\mathbf{s}^{[k+1]}) + LC(\mathbf{s}^{[k+1]} + (0, \ldots, 0, a)) \\ &= LC(\mathbf{s}^{[k+1]}) + LC(-\mathbf{s}^{[k+1]} - (0, \ldots, 0, a)) \\ &\geq LC(\mathbf{s}^{[k+1]} - \mathbf{s}^{[k]} - (0, \ldots, 0, a)) \\ &= LC((0, \ldots, 0, a)) = k+1. \end{aligned}$$

□

Con esto, podemos probar que el siguiente Algoritmo funciona:

Algoritmo 4.39 (de Berlekamp-Massey)**.** Dada $\mathbf{s}$ una sucesión en $\mathbb{F}_q$, el siguiente algoritmo genera secuencialmente, para cada $k \geq 0$, un polinomio mínimo $f^{[k]} \in \mathbb{F}_q[x]$ de $\mathbf{s}^{[k]}$.

1. Se busca el primer índice j tal que $s_{j-1} \neq 0$.

   a) Si no existe dicho j, cualquier polinomio genera s y cualquier truncación de s. El algoritmo termina.

   b) Si existe dicho j, se inicializan

   $$\begin{aligned} m &= j-1, & L_m &= 0, & f^{[m]} &= 1, & e_m &= s_j, \\ k &= j, & L_k &= j, & f^{[k]} &= x^j. \end{aligned}$$

2. Sea

   $$f^{[k]} = a_{k,L_k} x^{L_k} + \ldots + a_{k,0},$$

   se calcula la siguiente evaluación de $f^{[k]}$ en la sucesión s,

   $$e_k = a_{k,L_k} s_k + \ldots + a_{k,0} s_{k-L_k}. \tag{4.7}$$

a) Si $e_k = 0$, entonces $f^{[k]}$ genera $\mathbf{s}^{[k+1]}$, y se definen

$$L_{k+1} = L_k, \qquad f^{[k+1]} = f^{[k]}.$$

b) Si $e_k \neq 0$ entonces $f^{[k]}$ no genera $\mathbf{s}^{[k+1]}$, y se definen

$$L_{k+1} = \text{máx}\{L_k, k+1-L_k\}, \tag{4.8}$$
$$f^{[k+1]} = x^{L_{k+1}-L_k} f^{[k]} - (e_k/e_m) x^{L_{k+1}-k+m-L_m} f^{[m]}. \tag{4.9}$$

Si $L_{k+1} > L_k$, además se actualizan

$$m = k, \qquad L_m = L_k.$$

3. Se actualiza $k = k+1$, y se vuelve al paso 2.

Demostración:
Basta probar las cuatro afirmaciones siguientes:

- El polinomio $f^{[k+1]}$ está bien definido:
  Sabemos que $L_{k+1} \geq L_k$, por lo que $L_{k+1} - L_k \geq 0$. Supongamos que $L_{k+1} = L_k$, en ese caso $k+1-L_k \leq L_k$, y sabemos que $L_k = m+1-L_m$ (m se actualiza solo cuando cambia el grado, y $L_k = L_{m+1}$), entonces sustituyendo $k \leq 1 + 2(m - L_m)$, por lo que $L_{k+1} - k + m - L_m \geq 0$. Supongamos ahora que $L_{k+1} = k + 1 - L_k$. Entonces sustituyendo obtenemos $L_{k+1} - k + m - L_m = 0$.

- El grado de $f^{[k+1]}$ es L_{k+1}:
  El primer sumando tiene grado $\deg(x^{L_{k+1}-L_k} f^{[k]}) = L_{k+1}$, y el segundo sumando tiene grado $\deg(x^{L_{k+1}-k+m-L_m} f^{[m]}) = L_{k+1} - k + m - L_m + L_m \leq L_{k+1} - 1$. Con esto, sabemos que $\deg(f^{[k+1]}) = L_{k+1}$.

- El polinomio $f^{[k+1]}$ genera $\mathbf{s}^{[k+1]}$:
  Es evidente por la forma de $f^{[k+1]}$: para $n \leq (k-1) - L_{k+1}$, cada sumando se anula por separado, para $n = (k-1) - L_{k+1}$, el término $-(e_k/e_m)$ hace que un sumando se cancele con el otro.

- El polinomio $f^{[k+1]}$ es un polinomio mínimo de $\mathbf{s}^{[k+1]}$:
  Como $\deg(f^{[k+1]}) = L_{k+1}$, concluimos por el Lema anterior que el grado del polinomio es mínimo.

□

Observación 4.40. Por simplicidad conceptual, el algoritmo se plantea como un bucle infinito sobre una sucesión $\mathbf{s}$ también infinita. Si se aplica a una sucesión truncada $\mathbf{s}^{[k]}$, el algoritmo proporciona un polinomio mínimo $f^{[k]}$ en $O(k^2)$ operaciones. Si $L_k \leq k/2$, entonces $f^{[k]}$ es único, si $L_k > k/2$, entonces $f^{[k]}$ tiene $2k - L_k$ grados de libertad. Esto se puede deducir de la Demostración anterior, pero quizás es más satisfactorio y natural (si bien más elaborado) el enfoque en [20], donde primer se analiza cómo debe ser un polinomio mínimo, y desde ahí se deduce el algoritmo de Berlekamp-Massey.

Observación 4.41. Si algún cifrado en flujo utilizara como keystream una sucesión periódica con complejidad lineal pequeña L, y se conocieran $2L$ letras del texto claro, se podría obtener de manera inmediata, con el algoritmo de Berlekamp-Massey, el LFSR que genera el resto de la

sucesión (que no necesariamente debe provenir de un LFSR dentro del cifrado). En particular, esta ataque es particularmente serio frente a las m-sucesiones. Como vimos, estas sucesiones permitían un sencillo tratamiento matemático que probaba numerosas buenas propiedades, pero quizás el precio a pagar es que también admiten un sencillo criptoanálisis con el algoritmo de Berlekamp-Massey.

La idea detrás del algoritmo de Berlekamp-Massey es "ir arrastrando" $f^{[k]}$ por la sucesión, mientras se conserva el anterior $f^{[m]}$ en la posición en la que dejó de funcionar. Entonces, cada vez que falle $f^{[k]}$, lo combinamos con el $f^{[m]}$ (ambas "plantillas" vienen multiplicadas por una potencia de x adecuada, para ajustar su posición a la que corresponde realmente sobre la sucesión), haciendo que ambos errores se cancelen entre ellos. Realmente, todo el peso de la demostración del Algoritmo 4.39 se encuentra en el Lema 4.38, el resto son ajustes de índices. Quizás el mejor modo de entender esto es realizando a mano un par de ejemplos.

Ejemplo 4.42. Aplicamos el algoritmo de Berlekamp-Massey al tercer LFSR del Ejemplo 4.27 (cuyo polinomio es $x^6 + x^5 + x^3 + x + 1$), para las condiciones iniciales $(1, 0, 1, 0, 1, 0)$ y $(1, 1, 0, 1, 1, 0)$. Vemos cómo el algoritmo de Berlekamp-Massey obtiene los polinomios mínimos de las sucesiones truncadas $\mathbf{s}^{[k]}$, hasta llegar al algoritmo mínimo de la sucesión $\mathbf{s}$ (aunque en la primera tabla no se llega a considerar todo el periodo de $\mathbf{s}$; nosotros tenemos información privilegiada, pues ya conocemos el polinomio del LFSR). En particular, este polinomio mínimo no tiene porqué coincidir con el polinomio del LFSR, como vemos.
Como estamos operando en $\mathbb{F}_2$, las evaluaciones no nulas e_m, e_k siempre serán 1, por lo que se omiten en las tablas.

m	L_m	f^{[m]}	k	L_k	f^{[k]}	s_{k-1}
-	-	-	-	-	-	-
-	-	-	-	-	-	0
1	0	1	2	2	x^2	1
1	0	1	3	2	x^2	0
1	0	1	4	2	x^2+1	1
1	0	1	5	2	x^2+1	0
1	0	1	6	2	x^2+1	1
6	2	x^2+1	7	5	x^5+x^4+x^3	1
6	2	x^2+1	8	5	x^5+x^4+x^3	0
6	2	x^2+1	9	5	x^5+x^4+1	0
6	2	x^2+1	10	5	x^5+x^4+x^2+x+1	0
9	5	x^5+x^4+x^2+x+1	11	6	x^6+x^5+x^3+x+1	1
9	5	x^5+x^4+x^2+x+1	12	6	x^6+x^5+x^3+x+1	1
9	5	x^5+x^4+x^2+x+1	13	6	x^6+x^5+x^3+x+1	0

m	L_m	f^{[m]}	k	L_k	f^{[k]}	s_{k-1}
-	-	-	-	-	-	-
0	0	1	1	1	x	1
0	0	1	2	1	x+1	1
2	1	x+1	3	2	x^2+x+1	0
2	1	x+1	4	2	x^2+x+1	1
2	1	x+1	5	2	x^2+x+1	1
2	1	x+1	6	2	x^2+x+1	0
2	1	x+1	7	2	x^2+x+1	1
2	1	x+1	8	2	x^2+x+1	1
2	1	x+1	9	2	x^2+x+1	0
2	1	x+1	10	2	x^2+x+1	1
2	1	x+1	11	2	x^2+x+1	1
2	1	x+1	12	2	x^2+x+1	0
2	1	x+1	13	2	x^2+x+1	1

Figura 4.7: Dos tablas donde se han realizado paso a paso los cálculos del algoritmo de Berlekamp-Massey.

Nota 4.43. Dada una sucesión $\mathbf{s}$ en $\mathbb{F}_q$, en algunos textos se define su "perfil de complejidad lineal" como la sucesión de complejidades lineales de los sucesivos truncamientos $\mathbf{s}^{[k]}$. Como la Demostración del Algoritmo 4.39 nos dice que la desigualdad del Lema 4.38 es una igualdad, los saltos en el perfil de complejidad lineal son simétricos respecto a la recta $f(k) = k/2$.

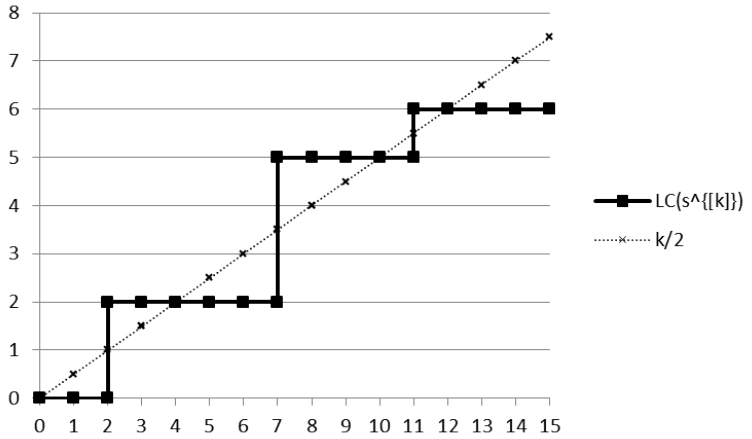

Figura 4.8: Perfil de complejidad lineal para la primera sucesión del Ejemplo anterior. Se ha extendido ligeramente el rango de valores, para mostrar que la simetría no es perfecta: si la sucesión es periódica, eventualmente, habrá un último salto en la complejidad lineal.

Esto proporciona otro posible test estadístico a una sucesión aleatoria. Nosotros no comentaremos más sobre el asunto, pero se puede encontrar un análisis detallado en [16][Capítulo 18].

El algoritmo de Berlekamp-Massey es el último tema que trataremos sobre LFSRs. Justamente, este algoritmo nos dice que no es viable utilizar un solo LFSR directamente en Criptografía. Pero ya habíamos adelantado que el papel de los LFSRs era ser "ladrillos" en criptosistemas más elaborado. ¿Cómo exactamente se deben colocar los ladrillos? Este será el tema del siguiente y último Capítulo.

Capítulo 5

Aplicaciones de los LFSRs en Criptografía

> Few false ideas have more firmly gripped the minds of so many intelligent men than the one that, if they just tried, they could invent a cipher that no one could break.
>
> *David Kahn*

La existencia del algoritmo de Berlekamp-Massey significa que debemos requerir una gran complejidad lineal para cualquier sucesión pseudoaleatoria que se vaya a usar en Criptografía (incluso si esta no viene generada por LFSRs). Presentaremos varios esquemas estándar basados en LFSRs que permiten generar sucesiones pseudoaleatorias con complejidades lineales mucho mayores que las de los LFSRs componentes.
Aún así, estos esquemas, por si solos, no son lo suficientemente seguros para su uso en Criptografía. Para lograr un cifrado seguro es necesario combinar varios de los esquemas, añadiendo y cambiando piezas hasta que el conjunto sea resistente a todos los ataques conocidos. Nosotros no podemos adentrarnos tanto en este "arte" de la Criptografía, pero si describiremos brevemente los cifrados en flujo de uso actual que se basan en LFSRs, que será un final satisfactorio para este Trabajo de Fin de Grado.

5.1. Combinadores no lineales

Una idea aparentemente razonable sería, en vez de considerar un solo LFSR, sumar la salida de varios. Desafortunadamente, a raíz de la Proposición 4.12, no se gana demasiado: la complejidad lineal de la suma será a lo sumo la suma de las combinaciones lineales. Es necesario pues emplear una función no lineal para combinar los LFSRs.

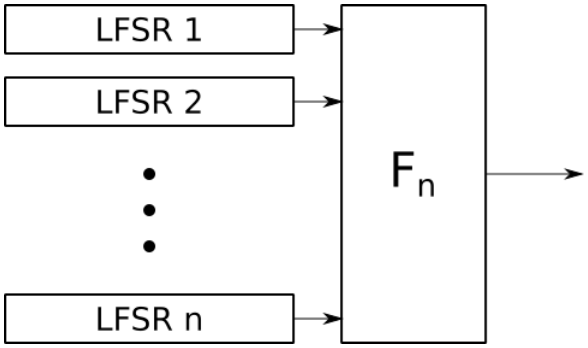

Figura 5.1: Diagrama de un combinador no lineal para n LFSRs.

Cuando tratamos con LFSRs con polinomios primitivos, tenemos el siguiente estético resultado.

Teorema 5.1. *Sean* $\mathbf{s}^1, \mathbf{s}^2, \ldots, \mathbf{s}^n$ *sucesiones recursivas lineales en* $\mathbb{F}_q$ *con complejidades lineales* L_i*. Si se combinan para formar la sucesión* $\mathbf{s}$*, con*

$$s_i = F_n(s_i^1, s_i^2, \ldots, s_i^n)$$

para todo $i \geq 0$*, donde* $F_n \colon \mathbb{F}_q^n \to \mathbb{F}_q$ *tiene la forma*

$$F_n(x_1, x_2, \ldots, x_n) = a_0 + \sum_{1 \leq i \leq n} a_i x_i + \sum_{1 \leq i,j \leq n} a_{i,j} x_i x_j + \ldots + a_{1,2,\ldots,n} x_1 x_2 \ldots x_n, \tag{5.1}$$

entonces, la complejidad linear de $\mathbf{s}$ *es a lo sumo* $F_n(L_1, L_2, \ldots, L_n)$*, donde* F_n *se ve ahora como una aplicación de* $\mathbb{Z}^n$ *en* $\mathbb{Z}$*. Si además las* $\mathbf{s}_i$ *son m-sucesiones, y los* L_i *son todos distintos y mayores que* 2*, entonces se alcanza la cota.*

Demostración:
La cota es una consecuencia directa de la Proposición 4.12 y la Proposición 4.13. Para una prueba de que se da la igualdad bajo las condiciones citadas, ver [33]. □

Observación 5.2. En particular, cuando trabajamos en $\mathbb{F}_2$, cualquier función $f \colon \mathbb{F}_2^n \to \mathbb{F}_2$ tiene la forma 5.1, ya que se puede escribir como

$$f(x_1, x_2, \ldots, x_n) = \sum_{y \in \mathbb{F}_2^n} F_n(y_1, y_2, \ldots, y_n)(x_1 - y_1)(x_2 - y_2) \ldots (x_n - y_n),$$

lo que se conoce como la forma normal algebraica. Entonces se define el orden no lineal de f como el grado de su forma algebraica, vista como polinomio en $\mathbb{F}_2[x_1, \ldots, x_n]$.

Observación 5.3. Este esquema también aumenta el periodo de la sucesión resultante. Si el LFSR i-ésimo tiene periodo T_i, el periodo total será $\text{lcm}(T_1, T_2, \ldots, T_n)$. Por esta razón, se suelen considerar LFSRs con longitudes coprimas.

Observación 5.4. Los combinadores no lineales, son vulnerables a los llamados "ataques de correlación". Estos ataques, que afectan a muchos tipos de cifrados, son en particular uno de los grandes responsables del fracaso de muchos cifrados en flujo basados en LFSRs. En consecuencia, merecen que les dediquemos unas pocas líneas.
Supongamos que tratamos con un combinador de 3 LFSRs en $\mathbb{F}_2$, cada uno con longitud 15, e imaginemos además que existe una correlación no nula entre el bit de salida b_i de cada LFSR y el bit de salida b_C del combinador: coinciden un 48 % de las ocasiones (frente a un 50 %, que sería lo deseable). Si poseemos los 2000 primeros bits del mensaje en texto claro (lo cuál no es tan raro, muchas veces la cabecera de los mensajes es estándar), podríamos ir probando candidatos a estado inicial (la clave) del primer LFSR, y para cada uno, generar los 2000 primeros bits y contar las coincidencias entre b_1 y b_C. Si la clave no es correcta, el número de coincidencias será aproximadamente 1000, si la clave es correcta, el número de coincidencias será aproximadamente 960. Se puede diseñar un test estadístico para distinguir ambos casos . Por ejemplo, si consideramos solo las claves con menos de 980 coincidencias, se calcula con el Teorema Central del Límite que descartaremos más del $99,4\,\%$ de las claves, mientras que la probabilidad de que rechacemos la clave correcta será de tan solo un $0,6\,\%$. Iterando este procedimiento para los 3 LFSRs, nos quedaríamos para cada uno con un pequeño número de claves candidatas, que iríamos probando en los siguientes. En total, con una probabilidad de éxito mayor que $98,8\,\% \approx (1 - 0,006)^2$, tendremos que probar unas $2^15 + (0,006)(2^15)^2 +$

$(0,006)^2(2^15)^3 \approx 1,27 \cdot 10^9$ claves, una mejora sustancial frente a las $10^45 \approx 3,52 \cdot 10^13$ que habría que probar en una búsqueda por fuerza bruta.
En resumen, lo que estamos haciendo es una búsqueda exhaustiva, pero en el que hemos aislamos parcialmente a cada LFSR de sus compañeros. De manera más general, si tenemos n LFSRs, cada uno de longitud L_i, y suficiente texto claro conocido como para diseñar un test estadístico con probabilidad de error prácticamente nula, los ataques de correlación reducen el número de claves a probar de $2^{L_1+L_2+\ldots+L_n}$ a $2^{L_1}+2^{L_2}+\ldots+2^{L_n}$, disminuyendo drásticamente la seguridad del esquema.
Para combatir esto, se piden funciones con lo que se llama "inmunidad a la correlación". Si consideramos la salida de los LFSRs como funciones aleatorias X_i uniformemente distribuidas en $\mathbb{F}_q$, se dice que F_n es inmune a la correlación hasta orden k, con $1 < k \leq n$, si $F_n(X_1, X_2, \ldots, X_n)$ es independiente de cualquier vector aleatorio formado por k de las n variables. Por otro lado, se dice que f está equilibrada si $P(F_n(X_1, X_2, \ldots, X_n) = 0) = P(F_n(X_1, X_2, \ldots, X_n) = 1)$.
Por desgracia, no siempre es posible lograr buena inmunidad a la correlación sin comprometer otros aspectos. En particular, cuando trabajamos en $\mathbb{F}_2$, se puede probar que si F_n es inmune a la correlación hasta orden k, entonces su orden no lineal es a lo sumo $n-k$, y si se pide además que f esté equilibrada, entonces su orden no linel es a lo sumo $n-k-1$ [36].

5.2. Filtrado no lineal

Otra opción es aplicar un filtro F en cada paso del LFSR: la sucesión de salida será una función del estado interno del LFSR. De nuevo, el filtro debe ser no lineal: si nos limitamos a sumar varias entradas, lo que obtenemos es la suma de varios desplazamientos de la sucesión de salida, que también es generada por el LFSR (lo que ya vimos en la Proposición 4.4).

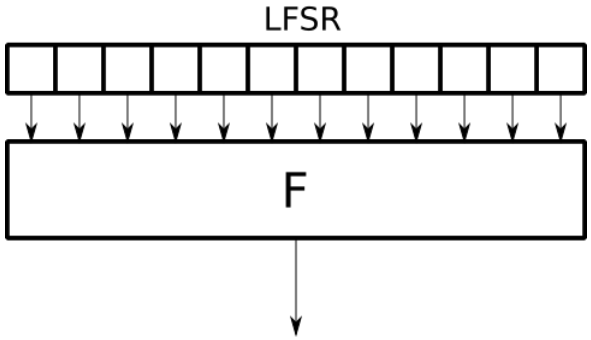

Figura 5.2: Diagrama de un LFSRs con un filtro no lineal.

En el caso particular de LFSRs binarios con polinomio primitivo, tenemos algunos resultados interesantes:

Teorema 5.5. *Sean* $\mathbf{s}$ *una m-sucesión en* $\mathbb{F}_2$ *de orden* k *y con complejidad lineal* L*. Si se aplica la función* $F\colon \mathbb{F}_q^n \to \mathbb{F}_q$ *para definir una nueva sucesión* $\mathbf{s}'$*, con*

$$s'_i = F(s_i, s_{i-1}, \ldots, s_{i-k}),$$

entonces la complejidad lineal de $\mathbf{s}'$ *es, a lo sumo,* $\sum_{j=1}^{m} \binom{L}{j}$*. Si* L *es primo, esta cota se alcanza para al menos una fracción* $\lceil e^{-1/L} \rceil$ *de los polinomios* F *de grado* k*. Si* F *es una función "doblada", es decir, tal que* $|\widehat{F}| \equiv cte.$*, entonces la complejidad lineal de* $\mathbf{s}'$ *será al menos* $2^{n/4}\binom{n/2}{n/4}$.

Demostración:
Para los dos primeros resultados, ver [12][sección 7.4]. Para el último resultado, ver [24]. □

Observación 5.6. Si bien este esquema requiere menos recursos que el anterior, por si solo, no aumenta el periodo, ya que el periodo de la sucesión recursiva lineal vectorial es el mismo que el periodo de la sucesión "escalar".

5.3. Otros esquemas

Los filtros y combinadores no lineales son quizás los esquemas más empleados para los que existen resultados generales. En esta sección mencionaremos brevemente otros esquemas importantes.

1. **Ciclado controlado**: Se emplea la salida de un LFSR, pero controlando su ciclado, es decir, no siempre devolviendo el siguiente elemento de la sucesión. La sucesión resultante se dice que es una "decimación" de la sucesión original. Distintos resultados, como por ejemplo en [16][Capítulo 10], nos indican que no es seguro eliminar términos de forma uniforme, sino que conviene emplear una fuente más pseudoaleatoria (como otro LFSR) para decidir qué elementos descartar. De esta forma, se introduce la necesaria no linealidad.
   Existen dos clases principales de generadores: los de paso alternado y los (auto)contraídos. Los generadores de paso alternado combinan varios LFSR con un simple XOR. La diferencia aquí es que no todos los LFRSs avanzan simultáneamente. Existe un LFSR principal que cicla cada pulso de reloj, y es el valor de este LFSR lo que controla cuál de los LFSRs secundarios avanza.

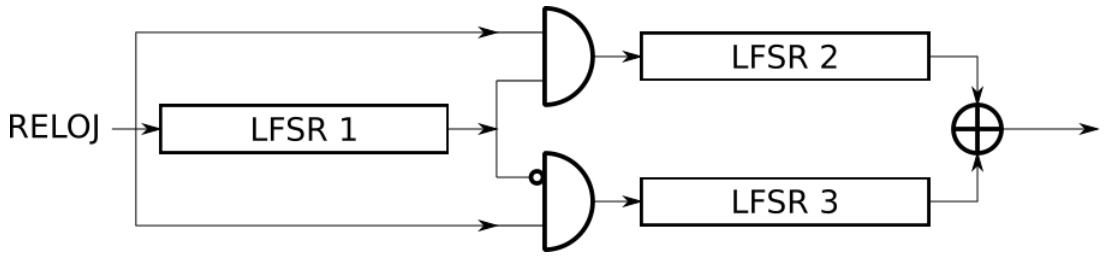

Figura 5.3: Diagrama de un generador de pulso alternado.

   Por otro lado, en un generador contraído, existen dos LFSRs, uno principal y otro auxiliar. Ambos avanzan cada pulso de reloj, pero el único LFSR que contribuye a la salida es el principal, mientras que el LFSR auxiliar decide si el nuevo elemento del LFSR principal se emitirá o no. Existe también la posibilidad de tomar los elementos de un solo LFSR de dos en dos, y hacer como si uno corresponde a un LFSR principal y el otro a un LFSR auxiliar. En ese caso, tenemos un generador "autocontraído".

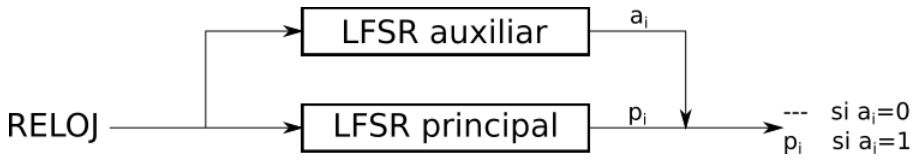

Figura 5.4: Diagrama de un generador contraído.

Una opción alternativa bastante empleada es establecer un "sistema de votación". Se consideran varios LFSRs, cada uno con una celda del estado interno marcada. En cada pulso de reloj, solo los LFSRs que contengan el símbolo mayoritario en su celda marcada avanzan.

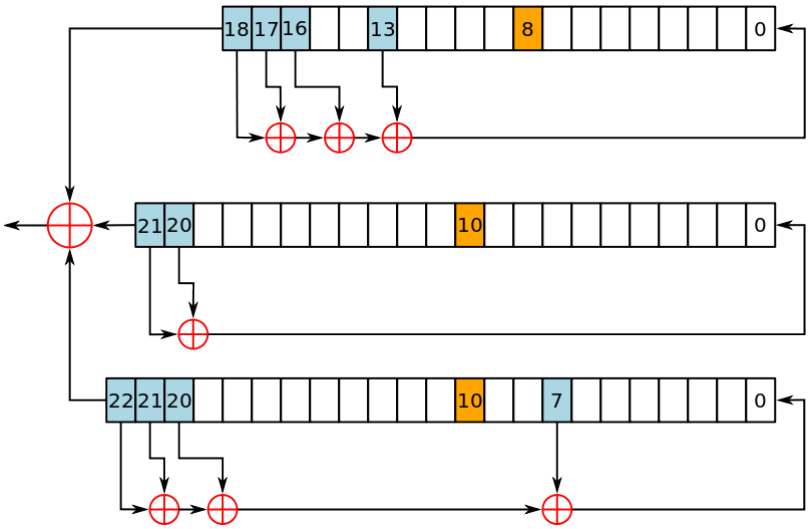

Figura 5.5: Diagrama del cifrado en flujo A5/1. En cada pulso de reloj, avanzan los LFSRs que tienen el bit naranja mayoritario. El ciclado conjunto es muy irregular, y sin embargo, las buenas propiedades de las m-sucesiones garantizan que ningún LFSR se puede "atascar" durante mucho tiempo. Este cifrado en flujo se usaba en 2G, pero ahora se considera vulnerable después de que aparecieran multitud de ataques serios.

2. **Distintas configuraciones de retroalimentación**: Ya vimos la configuración de Fibonacci y la configuración de Galois para un LFSR en la Observación 4.6. Sin duda, existen muchas otras formas de optimizar las conexiones sin cambiar la sucesión resultante [5][30]. Si bien no se logra solucionar el problema de la complejidad lineal, mejorar la eficiencia de los LFSRs permite dejar "hueco" para otros mecanismos que se encarguen de la no linealidad.

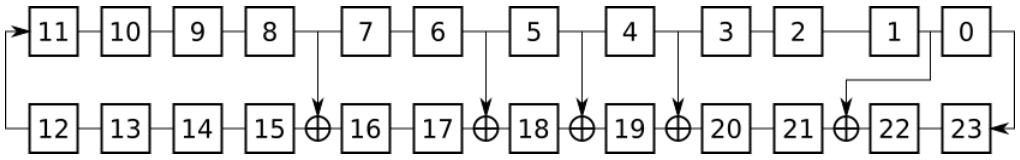

Figura 5.6: Diagrama de un LFSR con forma de anillo correspondiente al polinomio $x^{24} + x^{21} + x^{16} + x^{14} + x^{12} + x^8 + 1$ (esto viene explicado en [30]).

3. **Memorias**: Otro modo de complicar la salida de uno o varios LFSRs es introducir celdas de memoria. Estas celdas guardan temporalmente algún valor obtenido en pasos anteriores, lo que introduce una conexión entre cada paso y todos los anteriores, complicando el criptoanálisis.
   En particular, han sido muy analizados los llamados FCSR (Feedback With Carry Shift Registers), que funcionan como un LFSR, pero ahora en el anillo $\mathbb{Z}/(n)$ (si bien esto no es un cuerpo finito, la mayor parte de las propiedades que definimos para LFSRs en cuerpos finitos se pueden extender al caso de anillos, ver [16][Capítulo 3]). Como función de retroalimentación se considera la suma modular de varias celdas, la diferencia es que además nos "llevamos" el cociente módulo n a la celda de memoria, que pasa sumando al siguiente paso (al igual que el clásico algoritmo de suma que nos enseñan en la escuela). El estudio de los FCSR es muy interesante, en [16][Capítulo 4] se pueden encontrar una buena introducción a los FCSR con ayuda de números p-ádicos.

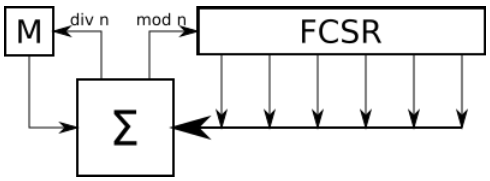

Figura 5.7: Diagrama de un FCSR.

De manera más general, es posible adjuntar a un LFSR una pequeña "máquina de estados finitos" (FSM, Finite State Machine), un pequeño subsistema con memorias que puede tomar un número limitado de estados, según el estado anterior, y la alimentación que reciba. Ambos sistemas se complementan, la salida regular pero poco uniforme del LFSR causa que la FSM evolucione de forma prácticamente impredecible.

4. **Cajas de sustitución**: Estas cajas son biyecciones precomputadas sobre un conjunto de valores dentro del circuito de cifrado. Suelen ser no lineales, específicamente diseñadas para conferir al cifrado resistencia a diversos ataques. Estas cajas son un elemento básico en las rondas de los cifrados en bloque, y de ahí algunos cifrados en flujo las han tomado prestadas.

5.4. Cifrados en flujo basados en LFSR con uso actual

Debido a su susceptibilidad a diversos ataques (que han causado algunos fracasos históricos, como el de A5/1), gran parte de los esquemas de cifrado en flujo basados en LFSRs han sido desfasado por otros. Por ejemplo, ya mencionamos que se utiliza ChaCha en el protocolo TLS, y sin duda la elección más popular es AES con algún modo de operación.
Aún así, los esquemas basados LFSRs siguen jugando un papel importante en la seguridad los sistemas de comunicación actuales. Aquí describiremos por encima tres de los más usados. Recordemos que, en todos los esquemas, la clave es el estado inicial de los LFSRs y las celdas de memoria (o más exactamente, estos se inicializan de forma determinista a partir de la clave, pero tampoco entraremos en exactamente cómo se supone que se debe inicializar cada cifrado).
El primer ejemplo que consideraremos es el cifrado en flujo E0, que es el cifrado usado por el protocolo Bluetooth (junto a AES-CTR, que si bien se ha introducido en las últimas versiones, muchos sistemas no lo soportan). Este cifrado consta de cuatro LFSRs en $\mathbb{F}_2$ con polinomios primitivos:

$$\begin{aligned} f_1(x) &= x^{25} + x^{20} + x^{12} + x^8 + 1, \\ f_2(x) &= x^{31} + x^{24} + x^{16} + x^{12} + 1, \\ f_3(x) &= x^{33} + x^{28} + x^{24} + x^4 + 1, \\ f_4(x) &= x^{39} + x^{36} + x^{28} + x^4 + 1. \end{aligned}$$

Se suman con XOR la salida de todos los LFSRs, pero estos también se conectan a una FSM con dos celdas de memoria de dos bits cada una (representadas por z^{-1}). Esta FSM introduce no linealidad al considerar la suma de bits sobre $\mathbb{Z}$, en vez de sobre $\mathbb{F}_2$, y también introduce una dependencia temporal, el valor c_{t+1} depende de c_t y de c_{t-1}. Por último, Z_1 y Z_2 son biyecciones lineales en $\mathbb{F}_2^2$. Para más detalles, ver [17].

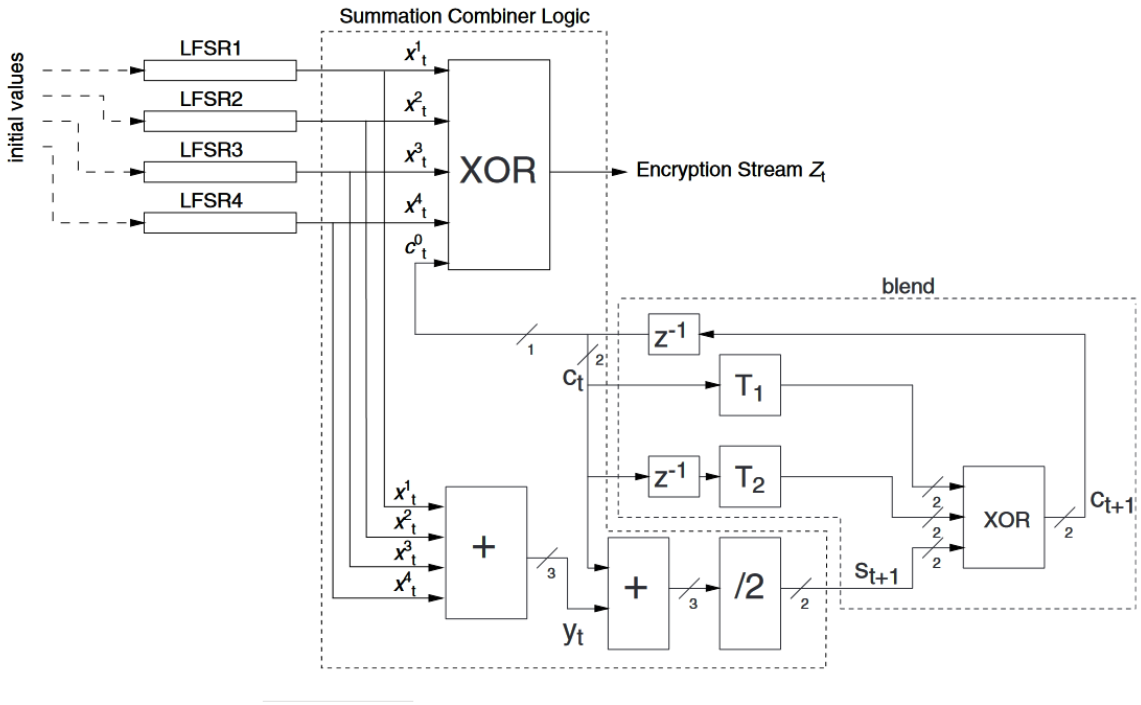

Figura 5.8: Diagrama del cifrado E0 en la especificación de Bluetooth [17].

Los otros dos ejemplos son los cifrados en flujo SNOW 3G y ZUC, ambos usados en los protocolos 4G y 5G [4]. Junto al ya viejo conocido AES-CTR, los tres cifrados se encuentran en situación de igualdad en el protocolo: cada aplicación/sistema conectado a 5G puede utilizar uno u otro según convenga.
En el cifrado en flujo SNOW 3G, primero se define un elemento $\alpha \in \mathbb{F}_{2^{32}}$, y a partir de él se define el LFSR en $\mathbb{F}_{2^{32}}$ con polinomio

$$f(x) = x^{16} - \alpha^{-1}x^{11} - x^2 - \alpha.$$

De este LFSR se extraen dos registros (que recordemos, cada uno contiene 32 bits), y se introducen en una FSM. Esta FSM contiene sumas en $\mathbb{F}_{2^{32}}$ (simbolizadas con un $\boxplus$) y operaciones XOR bit-a-bit (sumas en $(\mathbb{F}_2)^{32}$, simbolizadas por $\oplus$, que no son lineales en $\mathbb{F}_{2^{32}}$). Además, la FSM contiene tres celdas de memoria, conectadas por dos cajas de sustitución. Para más detalles, ver [3].

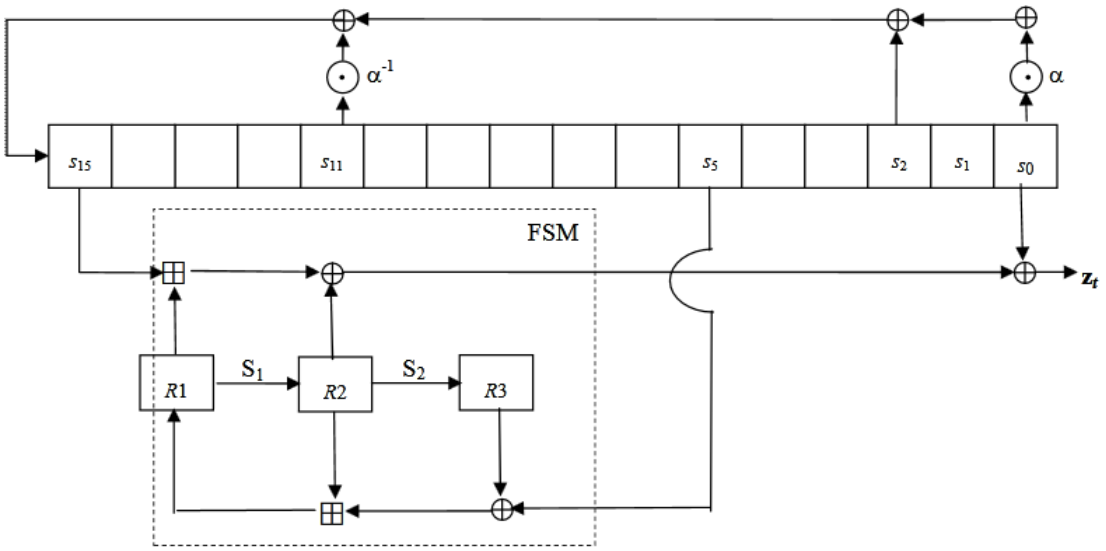

Figura 5.9: Diagrama del cifrado SNOW 3G en la especificación de ETSI/SAGE [3].

El cifrado en flujo ZUC también emplea un LFSR, pero ahora en $\mathbb{Z}/(2^{32})$, con polinomio

$$f(x) = x^{16} - 2^{15}x^{15} - 2^{17}x^{13} - 2^{21}x^{10} - 2^{20}x^4 - (1 + 2^8).$$

Aquí también se extraen elementos de 32 bits, en este caso recombinando varias mitades de 16 bits de los registros. Los tres primeros van a parar a una FSM con dos celdas de memoria R_1 y R_2, una suma lineal y una suma no lineal, un desplazamiento de 16 bits en un elemento concatenado de 64 bits (mezcla las mitades de los dos elementos de 32 bits), y dos cajas de sustitución S. Finalmente, se hace XOR bit-a-bit entre la salida de la FSM y el cuatro elemento de 32 bits que se obtuvo del LFSR. Para más detalles, ver [2].

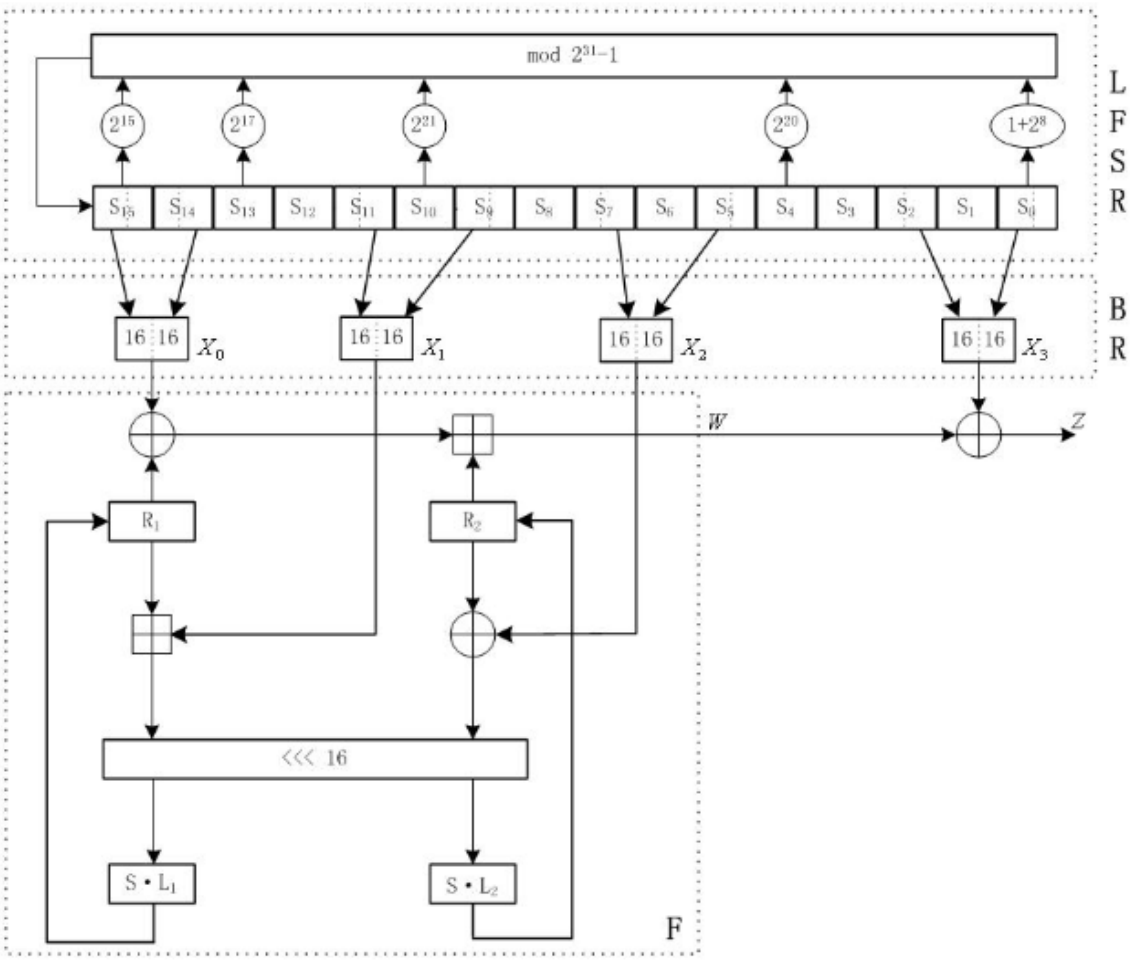

Figura 5.10: Diagrama del cifrado ZUC en la especificación de ETSI/SAGE [2].

Con el advenimiento de la Era de la Información, el número de aparatos conectados a la red crece exponencialmente, lo que requiere un cifrado en flujo extremadamente ligero. Incluso hoy en día, muchas instituciones buscan un nuevo cifrado que pueda implementarse con hardware mínimo, pero aún así sea seguro. Con ese objetivo se han convocado diversas competiciones, en particular, cabe destacar el proyecto *Lightweight Cryptography* del NIST. Varios candidatos de este concurso, como Grain, Elephant y Wage, contienen LFSRs como componentes principales, mientras que muchos otros utilizan pequeños LFSRs auxiliares para generar nuevas constantes en cada ronda de cifrado.

También es interesante e inexplorado el campo de los "NLFSRs", que son LFSRs en los que la función de retroalimentación ya no es lineal. Existen algunos esquemas para buscar NLFSRs que generen sucesiones con buenas propiedades (por ejemplo, [13]), pero, en general, la teoría detrás está poco desarrollada.

Sin duda, todavía queda mucho por estudiar sobre en el estudio de la Criptografía y los LFSRs. Por lo que a nosotros respecta, aquí finaliza este Trabajo de Fin de Grado.

Apéndice A

Teoría de cuerpos

Este Apéndice se plantea como una referencia de los resultados básicos de teoría de cuerpos que necesitaremos a lo largo de este Trabajo de Fin de Grado. La presentación está basada en [26][Capítulos 11-14], si bien se han realizado múltiples modificaciones.
Se ha procurado que el desarrollo sea autocontenido, solo se presupone una familiaridad muy superficial con la teoría de grupos y de anillos. No se pretende extender este Apéndice más de lo estrictamente necesario, así que inevitablemente dejaremos de lado muchos aspectos fundamentales de la teoría de cuerpos. Tampoco se plantearán ejemplos o se motivará en exceso las definiciones y resultados.

A.1. Extensión de cuerpos

Los cuerpos más comunes en Matemáticas son $\mathbb{Q}$, $\mathbb{R}$, y $\mathbb{C}$, y existe la relación de contención natural $\mathbb{Q} \subset \mathbb{R} \subset \mathbb{C}$ que respeta las operaciones. Esto nos motiva a explorar las relaciones de contención entre cuerpos con mayor generalidad.

Definición A.1. Sean K, L cuerpos, se dice que L extiende a K si existe un homomorfismo de cuerpos $\Phi\colon K \to L$. Generalmente se suele exigir que K sea un subcuerpo de L (de forma que el homomorfismo sea la inclusión ι), pero esto no le resta generalidad a la definición. De hecho, a veces abusaremos de la notación y consideraremos $K \subset L$.
Notemos que L se puede ver como un K-espacio vectorial. En este caso, se llama grado de la extensión $[L : K]$ a la dimensión de L como K-espacio vectorial. Una extensión se dice que es finita si su grado es finito.
Se define también el grupo de automorfismos de la extensión L de K, $\mathcal{A}ut(L/K)$, como el grupo (bajo la operación de composición) de automorfismos en L que dejan fijo K.

Notación A.2. Cuando decimos que un homomorfismo φ deja fijo un conjunto C, queremos decir que actúa como la identidad en él: $\varphi(x) = x$ para todo $x \in C$. Esto es diferente a decir que φ deja invariante C, que significaría que $\varphi(C) = C$.

Proposición A.3. *Sean K, F, L cuerpos, de forma que F es una extensión finita de K, y L es una extensión finita de F. Entonces*

$$[L : K] = [L : F][F : K].$$

Demostración:
Por simplificar la notación, sea $n = [L : F]$, $m = [F : K]$. Existe una base $\{e_i\}_{i=1}^n$ de L como F-espacio vectorial, existe una base $\{f_j\}_{j=1}^m$ de F como K-espacio vectorial. El conjunto

$$\mathcal{B} = \{e_i f_j \mid 1 \leq i \leq n,\ 1 \leq j \leq m\}$$

tiene nm elementos, y es sencillo comprobar que es una base de L como K-espacio vectorial. □

Definición A.4. Sea L una extensión de K cuerpo, consideramos un elemento $\alpha \in L$. Se dice que α es algebraico sobre K si existe un polinomio no nulo $f \in K[x]$ tal que $f(\alpha) = 0$. De lo contrario, se dice que α es trascendente sobre K. Si todos los elementos de L son algebraicos sobre K, se dice que L es una extensión algebraica de K. De lo contrario, se dice que L es una extensión trascendente de K.

La siguiente Proposición muestra, entre otras cosas, que toda extensión finita es algebraica.

Proposición A.5. *Dada una extensión finita L de K cuerpo, para todo $\alpha \in L$ existe un polinomio mónico irreducible $m_\alpha \in K[x]$ tal que $m_\alpha(\alpha) = 0$, y $m_\alpha \mid f$ para cualquier otro $f \in K[x]$ con $f(\alpha) = 0$. Las condiciones sobre m_α nos indican que es único, se dice que m_α es el polinomio mínimo de α sobre K.*

Demostración:
Notemos que al ser K cuerpo, $K[x]$ es un dominio de ideales principales. En particular, el ideal $I = \{f \in K[x] \mid f(\alpha) = 0\}$ es principal, está generado por un único elemento $m_\alpha \in K[x]$. Además m_α debe ser irreducible, porque de lo contrario α sería raíz de uno de sus factores con grado estrictamente menor. Falta ver que $I \neq (0)$, es decir, existe algún polinomio no nulo $f \in K[x]$ con $f(\alpha) = 0$.
Para ello, basta considerar la sucesión de conjuntos $A_0 = \{1\}$, $A_1 = \{1, \alpha\}$, $A_2 = \{1, \alpha, \alpha^2\}$, ... Como L es un K-espacio vectorial de dimensión finita, eventualmente algún A_n con $n \leq [L : K]$ será linealmente dependiente sobre K. Esto significa que existen coeficientes a_0, a_1, ..., a_n en K tal que $a_0 + a_1\alpha + \ldots + a_n\alpha^n = 0$. Dicho de otra forma, el polinomio $f_n = a_0 + a_1x + \ldots + a_nx^n$, $f_n \in K[x]$, verifica $f(\alpha) = 0$.
Observemos que si consideramos el primer A_n linealmente dependiente, obtenemos el polinomio mínimo si tomamos $m_\alpha = a_n^{-1} f_n$. □

Uno de los modos más naturales de extender un anillo A o un cuerpo K es considerar, respectivamente, su anillo de polinomios $R[x]$ o su cuerpo de expresiones racionales $K(x)$. Surge la pregunta: ¿qué ocurrirá si, en vez de considerar una variable "muda" x, evaluamos las expresiones anteriores en un elemento α de una extensión L de K cuerpo?

Notación A.6. Sea L una extensión de K cuerpo, consideramos elementos $\alpha_1, \ldots, \alpha_n \in L$.

1. Se denota por $K[\alpha_1, \ldots, \alpha_n]$ al mínimo anillo que extiende a K y contiene a $\alpha_1, \ldots, \alpha_n$. Es fácil ver que $K[\alpha_1, \ldots, \alpha_n] = \{f(\alpha_1, \ldots, \alpha_n) \mid f \in K[x_1, \ldots, x_n]\}$.

2. Se denota por $K(\alpha_1, \ldots, \alpha_n)$ al mínimo cuerpo que extiende a K y contiene a $\alpha_1, \ldots, \alpha_n$. Es fácil ver que $K(\alpha_1, \ldots, \alpha_n) = \{f(\alpha_1, \ldots, \alpha_n) \mid f \in K(x_1, \ldots, x_n)\}$.

En particular, $K[\alpha_1, \ldots, \alpha_n] \subset K(\alpha_1, \ldots, \alpha_n)$ como subanillo.

Proposición A.7. *Sea L una extensión de K cuerpo, consideramos un elemento $\alpha \in L$. Se tiene que α es algebraico sobre K si y solo si $K[\alpha] = K(\alpha)$. En este caso, el grado de la extensión, $[K[\alpha] : K]$, coincide con el grado del polinomio mínimo m_α de α sobre K.*

Demostración:

$\Rightarrow$| Primero probaremos que si α es algebraico sobre K, la dimensión de $K[\alpha]$ como K-espacio vectorial es finita. Sea m_α el polinomio mínimo de α sobre K, y sea $n = \deg(m_\alpha)$. Consideramos $\mathcal{B} = \{1, \alpha, \ldots, \alpha^{n-1}\}$, vamos a probar que $\mathcal{B}$ es una base de

$K[\alpha]$ como K-espacio vectorial.
Sea $f(\alpha)$ con $f \in K[x]$ un elemento genérico de $K[\alpha]$. Como K es un cuerpo, $K[x]$ es un dominio euclídeo, por lo que existen $q, r \in K[x]$ tal que $f = qm_\alpha + r$ y $\deg(r) < n$. De esta forma, $f(\alpha) = q(\alpha)m_\alpha(\alpha) + r(\alpha) = q(\alpha) \cdot 0 + r(\alpha) = r(\alpha)$, y como $\deg(r) < n$, está claro que $r(\alpha) \in \langle 1, \alpha, \ldots, \alpha^{n-1} \rangle$. Por tanto $\mathcal{B}$ es sistema de generadores de $K[\alpha]$ sobre K.
Por otra parte, sean a_0, a_1, ..., $a_{n-1} \in K$ de forma que $a_0 + a_1\alpha + \ldots + a_{n-1}\alpha^{n-1} = 0$, o dicho de otra forma, el polinomio $f = a_0 + a_1x + \ldots + a_{n-1}x^{n-1}$, $f \in K[x]$, verifica $f(\alpha) = 0$. Pero $\deg(f) \leq n - 1 < n = \deg(m_\alpha)$, por lo que debe ser $f = 0$, es decir, $a_0 = a_1 = \ldots = a_{n-1} = 0$. Con esto se concluye que $\mathcal{B}$ es también linealmente independiente sobre K.
Ahora estamos en condiciones de probar que $K[\alpha]$ es un cuerpo (y en consecuencia, $K(\alpha) = K[\alpha]$). Sea $f(\alpha)$ con $f \in K[x]$ un elemento genérico no nulo de $K[\alpha]$, basta probar que $f(\alpha)$ admite inverso en $K[\alpha]$. Ya hemos probado que $K[\alpha]$ es un K-espacio vectorial de dimensión finita, así que podemos aplicar un razonamiento análogo al de la Demostración de la Proposición A.5, y obtener que existe $g \in K[x]$ tal que $g(f(\alpha)) = 0$. Escribimos $g = ax^m(xh - 1)$, donde $a \in K^*$ y $h \in K[x]$ y evaluamos en $f(\alpha)$, $0 = g(f(\alpha)) = a(f(\alpha))^m(f(\alpha)h(f(\alpha)) - 1)$. Como L es un cuerpo, $K[\alpha]$ es un dominio de integridad, así que necesariamente $f(\alpha)h(f(\alpha)) - 1 = 0$, por lo que $(f(\alpha))^{-1} = h(f(\alpha)) \in K[\alpha]$.

$\Leftarrow$ Si $K[\alpha] = K(\alpha)$, en particular $\alpha^{-1} \in K[\alpha]$, es decir, existe $f \in K[x]$ tal que $\alpha^{-1} = f(\alpha)$. Pero esto significa que $\alpha f(\alpha) - 1 = 0$, es decir, α es una raíz de $(xf - 1) \in K[x]$, por lo que α es algebraico sobre K.

□

Observación A.8. Obviamente, $K[\alpha_1, \alpha_2, \ldots, \alpha_n] = (\ldots((K[\alpha_1])[\alpha_2])\ldots)[\alpha_n]$, y así mismo $K(\alpha_1, \alpha_2, \ldots, \alpha_n) = (\ldots((K(\alpha_1))(\alpha_2))\ldots)(\alpha_n)$. Por tanto, la Proposición anterior se extiende a que $\alpha_1, \alpha_2, \ldots, \alpha_n$ son todos algebraicos sobre K si y solo si $K[\alpha_1, \alpha_2, \ldots, \alpha_n] = K(\alpha_1, \alpha_2, \ldots, \alpha_n)$ (pero en este caso es más difícil decir algo sobre el grado de la extensión).

Ahora nos planteamos si dado un cuerpo K, es posible realizar el proceso inverso. Es decir, partiendo de un polinomio $f \in K[x]$, encontrar una extensión L de K tal que f tenga una raíz en L. Basta considerar uno de los factores irreducibles de f, en ese sentido enunciamos el siguiente resultado.

Proposición A.9. *Sea K un cuerpo, sea $f \in K[x]$ un polinomio irreducible. Se verifica:*

1. *$K[x]/(f)$ es un cuerpo que extiende a K.*
2. *Existe una raíz α de f en $K[x]/(f)$.*
3. *Sea $\alpha \in L$ con $f(\alpha) = 0$, donde L es una extensión de K. Entonces:*
   a) *$K[\alpha]$ es un cuerpo que extiende a K.*
   b) *Existe una raíz α de f en $K[\alpha]$.*
   c) *$K[x]/(f)$ es isomorfo a $K[\alpha]$.*

Demostración:

1. Por ser K cuerpo, en particular, es un anillo conmutativo y con unidad, y en consecuencia $K[x]/(f)$ también es un anillo conmutativo y con unidad. Para ver que $K[x]/(f)$ es cuerpo, falta ver que todo elemento no nulo de $K[x]/(f)$ tiene inverso. Sea $\bar{g} \neq \bar{0}$

un elemento de $K[x]/(f)$ con representante $g \in K[x]$. Como f es irreducible y $f \nmid g$, resulta que $\gcd(f, g) = 1$, y por la identidad de Bezout, existen $a, b \in K[x]$ tal que $af + bg = 1$. Como $\overline{f} = \bar{0}$, esto significa que $\bar{b}$ es el inverso de $\bar{g}$ en $K[x]/(f)$.
Además

$$\begin{aligned} \iota \colon K &\to K[x]/(f) \\ a &\mapsto \bar{a} \end{aligned}$$

es una inmersión canónica de K en $K[x]/(f)$, por lo que $K[x]/(f)$ extiende a K.

2. Con la inmersión anterior, f ahora lo vemos como un polinomio en $(K[x]/(f))[x]$. Entonces $\bar{x} \in K[x]/(f)$ será una raíz, pues $f(\bar{x}) = \overline{f} = \bar{0}$.

3. Ya sabemos que tanto $K[x]/(f)$ como $K[\alpha]$ extienden a K. En $K[x]/(f)$, $\bar{x}$ es una raíz de f, mientras que en $K[\alpha]$, α es una raíz de f. Esto nos lleva a considerar el homomorfismo

$$\begin{aligned} \Phi \colon K[x]/(f) &\to K[\alpha] \\ \bar{g} = g(\bar{x}) &\mapsto g(\alpha) \end{aligned}$$

Como $f(\alpha) = 0$, $\Phi(\bar{g})$ no depende del representante, y también está claro que Φ respeta las operaciones. Por ser homomorfismo de cuerpos, Φ debe ser inyectivo, y por otra parte la sobreyectividad es inmediata.

□

Fijado un cuerpo K, y un polinomio $f \in K[x]$, nos preguntamos si es posible iterar el proceso anterior hasta que obtengamos una extensión L de K tal que f factorice completamente sobre L (es decir, f se escriba como producto de factores lineales en $L[x]$). En este caso, podemos decir que "todas las raíces de f están en L": efectivamente, si extendemos L, no puede aparecer ninguna raíz adicional del polinomio.

Definición A.10. Dado un cuerpo K, un polinomio $f \in K[x]$, y una extensión L de K, se dice que L es un cuerpo de descomposición de f sobre K si f factoriza completamente sobre L, pero f no factoriza completamente sobre ningún subcuerpo propio de L.

Para justificar la Definición, debemos probar que para cualquier polinomio $f \in K[x]$ existe un único cuerpo de descomposición de f sobre K. Para esto necesitaremos un Lema previo.

Lema A.11. *Sea K cuerpo, sea $f \in K[x]$ un polinomio, y sea L un cuerpo de descomposición de f sobre K. Si tenemos M una extensión de K a través del homomorfismo $\Phi \colon K \to M$, entonces Φ se puede extender a un homomorfismo $\Phi \colon L \to M$ si y solo si $\Phi(f)$ factoriza completamente sobre M.*

Demostración:

$\Rightarrow|$ Basta trasladar la factorización de f sobre L aplicando Φ.

$\Leftarrow|$ Procedemos por inducción sobre el grado de la extensión. Si $[L : K] = 1$, no hay nada que probar. Ahora supongamos que $[L : K] > 1$. Sea g un factor irreducible de f sobre K con $\deg(g) > 1$ (debe existir pues f no factoriza completamente sobre K). Por hipótesis, g factoriza completamente sobre L, y $\Phi(g)$ factoriza completamente sobre M. En estas condiciones, dada α una raíz de f en L, tenemos que $\Phi(\alpha)$ es una raíz de $\Phi(f)$ en M.

Está claro que Φ induce un homomorfismo (bien definido) $\tilde{\Phi}$ en los anillos cociente

$$\begin{aligned}\tilde{\Phi}\colon K[x]/(g) &\to (\Phi(K))[X]/(\Phi(g))\\ \overline{a_n}\overline{x}^n + \ldots + \overline{a_1}\overline{x} + \overline{a_0} = h(\overline{x}) &\mapsto H(\overline{X}) = \overline{\Phi(a_n)}\overline{X}^n + \ldots + \overline{\Phi(a_1)}\overline{X} + \overline{\Phi(a_0)},\end{aligned}$$

también consideramos los isomorfismos naturales

$$\begin{aligned}u\colon K[x]/(g) &\to K[\alpha]\\ h(\overline{x}) &\mapsto h(\alpha),\end{aligned}$$

$$\begin{aligned}v\colon (\Phi(K))[X]/(\Phi(g)) &\to (\Phi(K))[\Phi(\alpha)]\\ H(\overline{X}) &\mapsto H(\Phi(\alpha)),\end{aligned}$$

y la inclusión

$$\begin{aligned}\iota\colon (\Phi(F))[\Phi(\alpha)] &\to M\\ H(\Phi(\alpha)) &\mapsto H(\Phi(\alpha)).\end{aligned}$$

En estas condiciones, $\Phi' = \iota \circ v \circ \tilde{\Phi} \circ u^{-1}$ extiende Φ a un homomorfismo $\Phi'\colon K[\alpha] \to M$. Además, $[L : K] = [L : K(\alpha)][K(\alpha) : K]$, y como $[K(\alpha) : K] > 1$, $[L : K(\alpha)] < [L : K]$, lo que permite aplicar la hipótesis de inducción: se puede extender $\Phi'\colon K[\alpha] \to M$ a un homomorfismo $\Phi''\colon L \to M$.

□

Teorema A.12. *Dado un polinomio $f \in K[x]$, existe un único cuerpo de descomposición de f sobre K, salvo isomorfismo que deje fijo K.*

Demostración:
Para probar la existencia, procederemos por inducción sobre el grado de f. Si $\deg(f) = 1$, no hay nada que probar. Supongamos que $\deg(f) > 1$, sea g un factor irreducible de f. Sabemos que $K[x]/(g)$ es una extensión de K, ahí $\alpha = \bar{x}$ es una raíz de g, y en consecuencia, de f. Por tanto, f factoriza como $f = h(X - \alpha)$, donde $f, h \in (K[x]/(g))[X]$ y $\deg(h) < \deg(f)$. Entonces podemos aplicar la hipótesis de inducción: existe una extensión L de $K[x]/(g)$ de forma que h factoriza completamente sobre L, y en consecuencia, f también. Si $f = (x - \alpha)\ldots(x - \alpha_n)$ en $L[x]$, entonces $K[\alpha_1, \ldots, \alpha_n] = K(\alpha_1, \ldots, \alpha_n)$ será un cuerpo de descomposición de f sobre K.
Ahora sean L, M dos cuerpos de descomposición de f sobre K. Si aplicamos el Lema anterior a la inclusión $\iota\colon K \to M$, obtenemos un homomorfismo de cuerpos $\iota'\colon L \to M$ que deja fijo K. Puesto que f factoriza completamente sobre L, $\Phi(f)$ factoriza completamente sobre $\iota'(L)$ un subcuerpo de M, así que necesariamente $\iota'(L) = M$. Esto significa que ι' es sobreyectivo, y como todo homomorfismo de cuerpos es inyectivo, hemos acabado. □

A.2. Cuerpos finitos

Es bien conocido que los anillos $\mathbb{Z}/(p)$ son cuerpos si y solo si p es primo. Esto nos invita a investigar si hay más cuerpos finitos, y qué se puede decir sobre ellos. Más adelante, veremos que estas estructuras algebraicas juegan un papel fundamental en el desarrollo teórico.
Primero, recordemos la definición de característica de un anillo.

Definición A.13. Sea R un anillo, si existe algún m natural tal que $m = m \cdot 1 = 1 + \ldots (\text{m veces}) \ldots + 1 = 0$, se dice que R tiene característica m. De lo contrario, se dice que R tiene característica 0.

Proposición A.14. *Todo cuerpo finito tiene característica p primo y orden p^n para cierto n natural.*

Demostración:
Sea K nuestro cuerpo, sea $F = \{m \mid m \in \mathbb{Z}\}$ un subanillo de K. Por ser un subanillo de un cuerpo, F debe ser un dominio de integridad. Como $F \subset K$, F es finito, y por tanto debe tener orden p finito. Esto induce un isomorfismo de anillos natural

$$\begin{aligned} \varphi\colon F &\to \mathbb{Z}/(p) \\ m &\mapsto \overline{m}, \end{aligned}$$

lo que nos permite deducir que p debe ser primo, pues solo en ese caso $\mathbb{Z}/(p)$ es un dominio de integridad (y también es un cuerpo).
Con esto hemos probado además que K extiende a $\mathbb{Z}/(p)$. Sea $n = [K : \mathbb{Z}/(p)]$ el grado de la extensión, por definición esta es la dimensión de K como $(\mathbb{Z}/(p))$-espacio vectorial. Entonces K tiene p^n elementos, es decir, su orden es p^n. □

Observación A.15. En particular, la característica de un cuerpo finito divide a su orden. Como consecuencia, si un cuerpo tiene orden p^n para cierto p primo, necesariamente debe tener como característica ese mismo p. En particular, todo cuerpo con p^n elementos extiende a $\mathbb{Z}/(p)$. Es más, dado un cuerpo K de característica p primo, $\mathbb{Z}/(p)$ es el menor subcuerpo de K (al igual que si tuviéramos un cuerpo K' de característica 0, $\mathbb{Q}$ sería el menor subcuerpo de K'). En este caso se dice que $\mathbb{Z}/(p)$ es el cuerpo base de K (y de modo análogo, se diría que $\mathbb{Q}$ es el cuerpo base de K').

En la siguiente Proposición veremos que la clasificación de los cuerpos finitos es *ligeramente* más sencilla que la de los grupos simples finitos.

Proposición A.16. *Para cada n natural, existe un único cuerpo de orden $q = p^n$, salvo isomorfismo que deje fijo K. Denotaremos a este cuerpo por $\mathbb{F}_q$.*

Demostración:
Sea p primo, $q = p^n$. Tomamos el polinomio $f = x^{p^n} - x$, donde se considera $f \in (\mathbb{Z}/(p))[x]$. Como $f' = (p^n)x^{p^n-1} - 1 = -1$, tenemos que $\gcd(f, f') = 1$ (esto es la derivada algebraica, se define de forma estándar para un polinomio, es rutina probar que cumple la regla del producto). Por tanto, en cualquier cuerpo K que extienda a $\mathbb{Z}/(p)$ y sobre el que factorice completamente f, las p^n raíces de f en K son distintas.
Sea $\mathbb{F}_q$ el cuerpo de descomposición de f sobre $\mathbb{Z}/(p)$, y sea $A \subset \mathbb{F}_q$ el conjunto de las p^n raíces. Veamos que A es un cuerpo. Sean α, α_1 y α_2 raíces de f. Es fácil ver que

1. $(\alpha_1 + \alpha_2)^{p^n} = (\alpha_1^p + \alpha_2^p)^{p^{n-1}} = \alpha_1^{p^n} + p^n \alpha_1^{p^n-1}\alpha_2 + \ldots + p^n \alpha_1 \alpha_2^{p^n-1} + \alpha_2^{p^n} = \alpha_1^{p^n} + \alpha_2^{p^n} = \alpha_1 + \alpha_2$

2. $(-\alpha)^{p^n} = (-1)^{p^n}\alpha^{p^n} = -\alpha$
3. $(\alpha_1\alpha_2)^{p^n} = \alpha_1^{p^n}\alpha_2^{p^n} = \alpha_1\alpha_2$
4. $(1/\alpha)^{p^n} = 1/\alpha^{p^n} = 1/\alpha$

Esto nos dice que A es un subcuerpo de $\mathbb{F}_q$ sobre el que f factoriza completamente. Como $\mathbb{F}_q$ es el cuerpo de descomposición de f sobre $\mathbb{Z}/(p)$, entonces necesariamente $A = \mathbb{F}_q$.
Sea ahora $\mathbb{K}_q$ otro cuerpo de p^n elementos, ya sabemos que $\mathbb{K}_q$ extiende a $\mathbb{Z}/(p)$. Su grupo multiplicativo $(\mathbb{K}_q)^*$ tiene p^n-1 elementos, en particular, todo elemento satisface $\alpha^{p^n-1} = 1$. Esto significa que f factoriza completamente sobre $\mathbb{K}_q$, y como las p^n raíces son distintas, f no factoriza completamente sobre ningún subcuerpo propio de $\mathbb{K}_q$. Por tanto, $\mathbb{K}_q$ es un cuerpo de descomposición de f sobre $\mathbb{Z}/(p)$, y por la unicidad de los cuerpos de descomposición, hemos acabado. □

Introducimos ahora la función φ de Euler, que será necesaria para el desarrollo posterior.

Definición A.17. Dado n natural, se define la función φ de Euler $\varphi(n)$ como la cantidad de naturales $m \leq n$ tal que $\gcd(m, n) = 1$.

El siguiente resultado es una de las propiedades importantes de la función φ de Euler.

Lema A.18. *Dados d, n naturales, en $\mathbb{Z}/(n)$ existen $\varphi(d)$ elementos con orden d.*

Demostración:
Por el Teorema de Lagrange, el orden d de cualquier elemento de este grupo debe dividir a el orden del grupo, que sabemos que es n. Fijado $d \mid n$, los elementos de orden d deben tener la forma $m(n/d)$, con $1 \leq m \leq d-1$ entero. Sea d' el orden de uno de estos elementos $m(n/d)$, como $m(n/d) = n(m/d)$, entonces $d' = d$ si y solo si $\gcd(m, d) = 1$. En consecuencia, $\varphi(d)$ es el número de elementos de $\mathbb{Z}/(n)$ con orden d. □

Observación A.19. Si en $\mathbb{Z}/(n)$ sumamos el número de elementos de cada orden d, como consecuencia del Lema anterior, obtenemos la "fórmula de Euler":

$$\sum_{d|n} \varphi(d) = \sum_{d|n} \varphi(n/d) = n.$$

Lema A.20. *Un grupo de orden n es cíclico si y solo si tiene exactamente $\varphi(d)$ elementos de orden d para todo $d \mid n$.*

Demostración:

$\Rightarrow$| Sea G nuestro grupo, sea $g \in G$ un elemento de orden d. El subgrupo $\langle g \rangle$ contiene a d elementos con orden que divide d, y por hipótesis, debe contener a todos los elementos de orden d' con $d' \mid d$. En particular, contiene a todos los elementos de orden d en G. Como $\langle g \rangle$ es isomorfo a $\mathbb{Z}/(d)$, podemos aplicar el Lema para concluir que si en G hay elementos de orden d, el número de estos elementos será $\varphi(d)$.

$\Leftarrow$| Esto se probó en el Lema A.18.

□

Ahora estamos en condiciones de analizar la estructura del grupo multiplicativo de todos los cuerpos finitos.

Proposición A.21. *El grupo multiplicativo $\mathbb{F}_q^*$ de $\mathbb{F}_q$ es un grupo cíclico (y en consecuencia, isomorfo a $\mathbb{Z}_{q-1}$). Si $\beta \in \mathbb{F}_q$ genera $\mathbb{F}_q^*$, diremos que β es un elemento primitivo de $\mathbb{F}_q$.*

Demostración:
Observamos que todos los elementos de orden d' con $d' \mid d$ en $\mathbb{F}_q^*$ son raíces de $x^d - 1$, de lo que se deduce que hay a lo sumo d de ellos. Entonces

$$q - 1 = |\mathbb{F}_q^*| \leq \sum_{d|(q-1)} \phi(d) = q - 1, \tag{A.1}$$

donde la primera igualdad resulta de contar todos los elementos de $\mathbb{F}_q^*$ según su orden, y la segunda es la fórmula de Euler. Esta expresión A.1 solo se puede verificar si la desigualdad $\leq$ es una igualdad $=$, lo que solo sucede cuando en $\mathbb{F}_q^*$ hay $\phi(d)$ elementos de orden d para todo $d \mid n$. Por el Lema anterior, concluimos que $\mathbb{F}_q^*$ es un grupo cíclico. □

El siguiente Lema es una herramienta básica, por ello será usado en múltiples ocasiones sin mención particular de aquí en adelante.

Lema A.22. *Sean m, n naturales. En cualquier anillo polinomial $R[x]$ son equivalentes:*

1. $m \mid n$
2. $x^m - 1 \mid x^n - 1$
3. $q^m - 1 \mid q^n - 1$

Demostración:

1. $\Rightarrow$ 2. Sea $n = bm$, entonces $x^n - 1 = (x^m - 1)(x^{(b-1)m} + x^{(b-2)m} + \ldots + x^m + 1)$, por lo que $x^m - 1 \mid x^n - 1$.

2. $\Rightarrow$ 3. Si $x^n - 1 = (x^m - 1)g(x)$, evaluando en q, $q^n - 1 = (q^m - 1)g(p)$, por lo que $q^m - 1 \mid q^n - 1$.

3. $\Rightarrow$ 1. Sea $n = bm + r$, donde $0 \leq r < m$. Efectuando la división de $q^n - 1$ entre $q^m - 1$, vemos que el resto tiene la forma $q^r - 1$. Pero esto es 0 si y solo si $r = 0$, por lo que $m \mid n$.

□

La siguiente Teorema recoge todas las relaciones de contención posibles entre cuerpos finitos.

Teorema A.23. *El cuerpo $\mathbb{F}_{q^n}$ extiende a $\mathbb{F}_{q^m}$ si y solo si $m \mid n$, y además lo hace de manera única (en el sentido de que todo un homomorfismo $\Phi\colon \mathbb{F}_{q^m} \to \mathbb{F}_{q^n}$ deja fijo $\mathbb{F}_{q^m}$, es decir, siempre se envía $\mathbb{F}_{q^m}$ al mismo subcuerpo de $\mathbb{F}_{q^n}$).*

Demostración:
Como

- todo homomorfismo fija los coeficientes del cuerpo base $\mathbb{Z}/(p)$ común a $\mathbb{F}_{q^m}$ y $\mathbb{F}_{q^n}$,
- el polinomio $x^{q^m-1} - 1$ tiene coeficientes en dicho cuerpo base,
- $\mathbb{F}_{q^m}$ es el cuerpo de descomposición de $x^{q^m-1} - 1$ sobre $\mathbb{Z}/(p)$,

podemos aplicar el Lema A.24: $\mathbb{F}_{q^n}$ extiende a $\mathbb{F}_{q^m}$ si y solo si $x^{q^m-1} - 1$ factoriza completamente sobre $\mathbb{F}_{q^n}$. Como $\mathbb{F}_{q^n}$ está formado por las $q^n - 1$ raíces distintas de $x^{q^n-1} - 1$ junto con

el 0, entonces $x^{q^m-1}-1$ factoriza completamente sobre $\mathbb{F}_{q^n}$ si y solo si $x^{q^m-1}-1 \mid x^{q^n-1}-1$, lo que sucede si y solo si $q^m-1 \mid q^n-1$, que a su vez es equivalente a $m \mid n$.
En este caso, el subcuerpo isomorfo a $\mathbb{F}_{q^m}$ debe ser $A = \{\alpha \in \mathbb{F}_{q^n} \mid \alpha^{q^m-1}-1=0\}$. □

La Proposición A.27 solo la requerimos en la forma enunciada, pero cabe destacar que se podría generalizar en el marco de las extensiones llamadas "normales". Pero nosotros solo la requeriremos en la forma enunciada. Antes necesitamos un par de pequeños Lemas.

Lema A.24. *Sea L una extensión de K un cuerpo, sea $f \in K[x]$ un polinomio, y sea α una raíz de f en L. Si L' es el cuerpo de descomposición de f sobre K, entonces L' extiende a $K[\alpha]$.*

Demostración:
Sea g un factor irreducible de f tal que $g(\alpha)=0$. Como f factoriza completamente sobre L', g también lo hace, sea α' una raíz de g en L'. Sabemos que L' extiende a $K[\alpha']$, y este cuerpo es isomorfo a $K[x]/(g)$, que es isomorfo a $K[\alpha]$. □

Definición A.25. Fijado $\mathbb{F}_q$, para cada $\mathbb{F}_{q^n}$ que extiende a $\mathbb{F}_q$, y para cada j natural, definimos los endomorfismos de cuerpos

$$\begin{aligned} \sigma_j \colon \mathbb{F}_{q^n} &\to \mathbb{F}_{q^n} \\ \alpha &\mapsto \alpha^{q^j}. \end{aligned}$$

También definimos, abusando de la notación, los endomorfismos de anillos

$$\begin{aligned} \sigma_j \colon \mathbb{F}_q[x] &\to \mathbb{F}_q[x] \\ a_n x^n + \ldots + a_0 = h(x) &\mapsto \sigma_j(h(x)) = a_n^{q^j}(x^n)^{q^j} + \ldots + (a_0)^{q^j}. \end{aligned}$$

Será evidente por el contexto cuándo nos referimos a uno u otro endomorfismo. De todas formas, no hay riesgo de confusión, ya que ambos actúan como uno esperaría que actuara σ_j en los subespacios pertinentes de $\mathbb{F}_{q^n}[x]$ si lo definiéramos allí.
En ambos casos, llamamos endomorfismo de Frobenius a σ_1. Este endomorfismo tiene especial importancia, ya que genera los otros endomorfismos por composición, $\sigma_j = (\sigma_1)^j$.

Lema A.26. *Se verifican las siguientes propiedades:*

1. *Los σ_j son efectivamente endomorfismos, y de hecho, son automorfismos en $\mathbb{F}_{q^n}$. Los σ_j son efectivamente endomorfismos en $\mathbb{F}_q[x]$.*
2. *Se tiene que $\sigma_j = \sigma_k$ en $\mathbb{F}_{q^n}$ si y solo si $j \equiv k \bmod n$. En particular, los σ_j dejan fijo $\mathbb{F}_q$, y $\sigma_n = id$.*
3. *Los σ_j forman un grupo cíclico $\langle \sigma_1 \rangle = \{\sigma_1, \ldots, \sigma_n\}$ con la operación de composición en $\mathbb{F}_{q^n}$.*
4. *Se tiene que $\sigma_j(h(x)) = h(\sigma_j(x))$ para todo $h \in \mathbb{F}_q[x]$.*

Demostración:

1. En ambos casos, la única dificultad para probar que los σ_j son endomorfismos es la linealidad de la suma. Esta se obtiene aplicando el binomio de Newton, teniendo en cuenta que q es una potencia de p y que tanto $\mathbb{F}_{q^n}$ como $\mathbb{F}_q$ tienen característica p. Queda notar que todo endomorfismo en un cuerpo finito es un automorfismo, ya que es inyectivo por ser homomorfismo de cuerpos, y debe ser sobreyectivo por ser una inyección de un conjunto finito en sí mismo.

2. Sea $k = bn + j$, recordemos que $\alpha^{q^n} = \alpha$ para todo $\alpha \in \mathbb{F}_{q^n}$. Operamos,

$$\begin{aligned}\alpha^{q^k} = \alpha^{q^{bn+j}} &= (\alpha^{q^n})^{q^{(b-1)n+j}} = (\alpha)^{q^{(b-1)n+j}} \\ &= \dots \\ &= (\alpha^{q^n})^{q^{(1-1)\cdot n+j}} = (\alpha)^{q^{0\cdot n+j}} = \alpha^{q^j}.\end{aligned}$$

   Para la otra implicación, basta considerar un elemento primitivo de $\mathbb{F}_{q^n}$.
   La prueba sigue funcionando si definimos σ_0 como $\sigma_0(\alpha) = \alpha^{q^0} = \alpha^1 = \alpha$, y de esta apreciación surgen los dos casos particulares.

3. Se deduce directamente de $\sigma_n = id$ en el punto anterior.

4. Sea $h \in \mathbb{F}_q[x]$, $h = a_n x^n + \ldots + a_1 x + a_0$. Como ya hemos probado que σ es endomorfismo en $\mathbb{F}_q[x]$, lo único que falta es notar que $(a_i)^{q^j} = a_i$, ya que σ_1 deja fijo $\mathbb{F}_q$.

□

Proposición A.27. *Sea $f \in \mathbb{F}_q[x]$ un polinomio irreducible de grado n, entonces f factoriza completamente sobre $\mathbb{F}_{q^n}$. Además, si α es una raíz de f en $\mathbb{F}_{q^n}$ es una de sus raíces, el resto de raíces son $\alpha^q, \alpha^{q^2}, \ldots, \alpha^{q^{n-1}}$.*

Demostración:
Sabemos que $\mathbb{F}_q[x]/(f)$ es una extensión de grado n de $\mathbb{F}_q$, y por la unicidad los cuerpos finitos, $\mathbb{F}_q[x]/(f)$ debe ser isomorfo a $\mathbb{F}_{q^n}$. En consecuencia f tiene una raíz $\alpha \in \mathbb{F}_{q^n}$. Consideramos los σ_j con $1 \leq j \leq n-1$, sabemos que $\sigma_j(f(x)) = f(\sigma_j(x))$, así que todos los $\alpha, \alpha^q, \alpha^{q^2}, \ldots, \alpha^{q^{n-1}}$ son raíces de f. Falta comprobar que no hay dos iguales.
De lo contrario, tendríamos $\alpha^{q^j} = \alpha^{q^k}$ con $1 \leq j < k \leq n-1$. Elevando repetidas veces a la q, esto nos dice que $\alpha^{q^{n-(k-j)}} = \alpha^{q^n} = \alpha$, lo que significa que α es una raíz de $x^{q^{n-k+j}} - x$. Como $\mathbb{F}_{q^{n-k+j}}$ es el cuerpo de descomposición de $x^{q^{n-k+j}} - x$ sobre $\mathbb{F}_q$, podemos aplicar el Lema A.11: $\mathbb{F}_{q^{n-k+j}}$ extiende a $\mathbb{F}_q[\alpha] = \mathbb{F}_{q^n}$. Pero $n - k + j < n$, absurdo. □

Dada L una extensión de K cuerpo, la teoría de Galois busca relaciones entre el conjunto de subcuerpos M de L que extienden a K, y el grupo de automorfismos $\mathcal{A}ut(L/K)$. Cuando la extensión es de un tipo particular llamado "de Galois", el grupo de automorfismos $\mathcal{A}ut(L/K)$ se llama grupo de Galois $\mathcal{G}al(L/K)$. Para las extensiones de Galois y finitas, se puede probar un bello resultado llamado "Teorema fundamental de la teoría de Galois", que determina completamente la estructura de la extensión. Desarrollar la teoría de Galois para extensiones de cuerpo generales no interesa demasiado en este Trabajo de Fin de Grado, cuyos objetivos son otros.
Pero sí diremos que, con lo que hemos probado hasta ahora, es inmediato ver que si L (y en consecuencia K) es un cuerpo finito, entonces la extensión es de Galois (y también finita). El siguiente Teorema es el caso particular del Teorema fundamental de la teoría de Galois cuando ambos cuerpos son finitos.

Teorema A.28. *El grupo de Galois de la extensión $\mathbb{F}_{q^n}$ de $\mathbb{F}_q$ es cíclico y está generado por el endomorfismo de Frobenius, es decir, $\mathcal{G}al(L/K) = \langle \sigma_1 \rangle = \{\sigma_1, \ldots, \sigma_n = id\}$.*

Demostración:
Ya sabemos que estos endomorfismos son automorfismos que dejan $\mathbb{F}_q$ fijo, y que forman un grupo cíclico bajo la composición que está generado por σ_1. Sea ahora σ' otro automorfismo en $\mathbb{F}_{q^n}$ que deja fijo $\mathbb{F}_q$. Consideramos β un elemento primitivo de $\mathbb{F}_{q^n}$, y sea m_β su polinomio

mínimo sobre $\mathbb{F}_q$. Si aplicamos σ' a $m_\beta(\beta) = 0$, tenemos $0 = \sigma'(m_\beta(\beta)) = m_\beta(\sigma'(\beta))$, lo que nos dice que $\sigma'(\beta)$ también es una raíz de m_β, y por la Proposición anterior deberá ser $\sigma(\beta) = \beta^{q^j}$ para cierto j con $0 \leq j \leq n-1$. Como β es un elemento primitivo de $\mathbb{F}_{q^n}$, todo elemento no nulo de $\mathbb{F}_{q^n}$ es una potencia de β, de lo que se deduce que $\sigma'(\alpha) = \alpha^{q^j}$ para todo $\alpha \in \mathbb{F}_{q^n}$, es decir, $\sigma' = \sigma_j$. □

Con esto, hemos concluido los prerrequisitos matemáticos. Sin duda, queda abierta la cuestión de *cómo* realizar cálculos en un cuerpo finito. Si bien nosotros no nos preocuparemos de ese tema (aunque es importante en el diseño de algunos de los cifrados en 5.4), dejamos una referencia [22].

Bibliografía

[1] Linear-feedback shift register. `https://en.wikipedia.org/wiki/Linear-feedback_shift_register`. Accessed: 2020-09-23.

[2] *Specification of the 3GPP Confidentiality and Integrity Algorithms 128-EEA3 & 128-EIA3. Document 2: ZUC Specification.* https://www.gsma.com/aboutus/wp-content/uploads/2014/12/eea3eia3zucv16.pdf, 1.6 edition.

[3] *Specification of the 3GPP Confidentiality and Integrity Algorithms UEA2 & UIA2. Document 2: SNOW 3G Specification.* https://www.gsma.com/aboutus/wp-content/uploads/2014/12/snow3gspec.pdf, 1.1 edition.

[4] 3GPP TS 33.501 (version 16.2.0). Technical report, 3GPP, 2019.

[5] F. Arnault, T. Berger, M. Minier, and B. Pousse. Revisiting lfsrs for cryptographic applications. *IEEE Transactions on Information Theory*, 57(12):8095–8113, 2011.

[6] Jean-Philippe Aumasson, Simon Fischer, Shahram Khazaei, Willi Meier, and Christian Rechberger. New Features of Latin Dances: Analysis of Salsa, ChaCha, and Rumba. In Kaisa Nyberg, editor, *Fast Software Encryption*, pages 470–488, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[7] Daniel J. Bernstein. Cache-timing attacks on AES. 2005.

[8] Andrey Bogdanov, Dmitry Khovratovich, and Christian Rechberger. Biclique cryptanalysis of the full AES. volume 7073, pages 344–371. Springer, 2011. 17th International Conference on the Theory and Application of Cryptology and Information Security, ASIACRYPT 2011 ; Conference date: 01-01-2011.

[9] David Brink. A (probably) exact solution to the Birthday Problem. *The Ramanujan Journal*, 28(2):223–238, apr 2012.

[10] Johannes A. Buchmann. *Introduction to Cryptography*. Springer New York, 2004.

[11] Carlos Iborra Castillo. *Representaciones de Grupos Finitos*. https://www.uv.es/ivorra/Libros/Representaciones.pdf.

[12] Thomas Cusick and Pante Stănică. *Cryptographic Boolean Functions and Applications*. Elsevier, 2009.

[13] Elena Dubrova, Maxim Teslenko, and Hannu Tenhunen. On analysis and synthesis of (n, k)-non-linear feedback shift registers. DATE '08, New York, NY, USA, 2008. Association for Computing Machinery.

[14] Morris J. Dworkin, Elaine B. Barker, James R. Nechvatal, James Foti, Lawrence E. Bassham, E. Roback, and James F. Dray Jr. Advanced Encryption Standard (AES). Technical report, National Institute of Standards and Technology, 2001.

[15] Solomon W Golomb. *Shift Register Sequences*. WORLD SCIENTIFIC, sep 2014.

[16] Mark Goresky and Andrew Klapper. *Algebraic Shift Register Sequences*. Cambridge University Press.

[17] Core Specification Working Group. Bluetooth Core Specification (version 5.2). Technical report, Bluetooth, 2019.

[18] Shay Gueron. Intel® Advanced Encryption Standard (AES) New Instructions Set. Technical report, Intel Corporation, 2010.

[19] Tor Helleseth. Golomb's last theorem. *9th Nordic Combinatorial Conference*, 2007.

[20] K. Imamura and W. Yoshida. A simple derivation of the berlekamp- massey algorithm and some applications (corresp.). *IEEE Transactions on Information Theory*, 33(1):146–150, 1987.

[21] Aditi Kar. Weyl's equidistribution theorem. *Resonance*, 8(5):30–37, 2003.

[22] John Kerl. Computation in finite fields, 2004.

[23] Andreas Klein. *Stream Ciphers*. Springer, 12 2013.

[24] P. V. Kumar and R. Scholtz. Bounds on the linear span of bent sequences. *IEEE Trans. Inf. Theory*, 29:854–862, 1983.

[25] Rudolf Lidl and Harald Niederreiter. *Introduction to finite fields and their applications*. Cambridge University Press, 1st edition, 1986.

[26] Rudolf Lidl and Günter Pilz. *Applied Abstract Algebra*. Springer, 2nd edition, 1998.

[27] David A. McGrew. Impossible plaintext cryptanalysis and probable-plaintext collision attacks of 64-bit block cipher modes. *IACR Cryptol. ePrint Arch.*, 2012:623, 2012.

[28] Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, dec 2018.

[29] N. David Mermin. *Quantum Computer Science*. Cambridge University Press, 2007.

[30] N. Mukherjee, J. Rajski, Grzegorz Mrugalski, Artur Pogiel, and J. Tyszer. Ring generator: An ultimate linear feedback shift register. *Computer*, 44:64–71, 2011.

[31] Kaisa Nyberg. Lecture notes for t-79.5501 cryptology (5 cr) l, 2008.

[32] Vladimir Rosenfeld. Enumerating de bruijn sequences. *MATCH Communications in Mathematical and in Computer Chemistry*, 01 2002.

[33] R. A. Rueppel and O. Staffelbach. Products of linear recurring sequences with maximum complexity. *IEEE Trans. Inf. Theory*, 33:124–131, 1986.

[34] Andrew Rukhin, Juan Soto, James Nechvatal, Miles Smid, Elaine Barker, Stefan Leigh, Mark Levenson, Mark Vangel, David Banks, Alan Heckert, James Dray, and San Vo. A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications. Technical report, NIST, 2010.

[35] C. E. Shannon. Communication Theory of Secrecy Systems. *Bell System Technical Journal*, 28(4):656–715, oct 1949.

[36] T. Siegenthaler. Correlation-immunity of nonlinear combining functions for cryptographic applications (corresp.). *IEEE Transactions on Information Theory*, 30(5):776–780, 1984.

[37] John Sullivan. Lecture notes for math 317, 2003.

[38] He Hans Wanders. On the significance of golomb's randomness postulates in cryptography. *Philips Journal of Research*, 43:185–222, 1988.

[39] Neal Zierler and WH Mills. Products of linear recurring sequences. *Journal of Algebra*, 27(1):147–157, 1973.