"ARITMÉTICA RÁPIDA SOBRE CAMPOS FINITOS"

TESIS QUE PRESENTA

JULIA IZTACCIHUALT MUÑOZ GUERRA

PARA LA OBTENCIÓN DEL GRADO DE

MAESTRA EN CIENCIAS (MATEMÁTICAS)

ASESOR: DR. HORACIO TAPIA RECILLAS

MAYO DEL 2001

UNIVERSIDAD AUTÓNOMA METROPOLITANA-IZTAPALAPA

DIVISIÓN DE CIENCIAS BÁSICAS E INGENIERÍA

"ARITMÉTICA RÁPIDA SOBRE CAMPOS FINITOS"

TESIS QUE PRESENTA

JULIA IZTACCIHUALT MUÑOZ GUERRA

PARA OBTENER EL GRADO DE
MAESTRÍA EN CIENCIAS (MATEMÁTICAS)

A mis Padres:

Pablo y Teresa

A mis hermanos:
Pedro Tlaloc
Tonancin e
Isabel Coyolxauhqui.

Agradecimientos

A mi familia: Pablo, Teresa, Pedro Tlaloc, Tonancin, Isabel Coyolxauhqui, Teresa, Anastacio, Gerardo Christian, Jenniria Asereth, Juan Pablo y Gabriela por todo su cariño y apoyo.

A los profesores: Dr. Horacio Tapia Recillas y a la Dra. Laura Hidalgo de la UAM-Iztapalapa y al Dr. Pedro Luis del Angel del CIMAT por aceptar ser mis sinodales.

A mis amigos y compañeros: Sandra Díaz Santiago, Jorge Alejandro Carrillo Ugalde, Vera Alicia Rodríguez Coronado, Juan Carlos Aguilar Franco, Joaquin Urbina, Veronica Pérez y Javier Pérez.

A los profesores del departamento de Matemáticas de la UAM-Iztapalapa: Dra. Lourdes Palacios, Dr. Carlos Signoret, Dr. Ernesto Pérez, Dra. María José Arroyo, Dra. Patricia Saavedra, Dr. Richard Wilson, M. en C. Adolfo Torres, Dr. Alfredo Nicolas y Dra. Blanca Rosa Pérez.

Deseo agregar que este trabajo fue financiado parcialmente por el proyecto "Teoría Algebraica de Códigos, Algebra Conmutativa y Geometría Algebraica"Convenio No. 400200-5-L00076-E9607. Bajo la responsabilidad del Dr. Horacio Tapia Recillas.

Finalmente al Departamento de Matemáticas por todas las facilidades que me otorgaron durante mis estudios de Maestría.

Índice General

Índice de Figuras

Índice de Tablas

Prefacio

El estudio de los campos finitos ha tenido un gran desarrollo en los últimos años debido a que se aplica en áreas como: *criptografía, teoría de códigos, combinatoria y geometría algebraica* entre otros. El ejemplo mas común (y prático) que todos conocemos es el de los binarios.

Al trabajar con campos finitos se hacen operaciones como suma, multiplicación y división. Por lo que se necesita encontrar la mejor representación y algoritmos eficientes para efectuar la aritmética. Siendo éste un problema de gran importancia actualmente.

Para mejorar los algoritmos utilizados en la aritmética sobre campos finitos, requerimos reunir conceptos tanto de las Matemáticas (propiedades de las bases de un campo finito) como conceptos de Computación (complejidad computacional y análisis de algoritmos).

Este trabajo trata principalmente sobre algunos métodos de multiplicación y exponenciación sobre campos finitos. Estos métodos dependen de la base usada (*bases polinomiales* o *bases normales*), puesto que las propiedades de las bases repercuten en la eficiencia de la aritmética, excepto el algoritmo de la suma ya que este método tiene la misma complejidad sin importar la base utilizada.

Partes de este trabajo, se han presentado en eventos nacionales como el 4^o Coloquio Nacional de Teoría de Códigos y Criptografía y Áreas Relacionadas, y en el XXXIII Congreso de la Sociedad Matemática Mexicana.

Introducción

En la Teoría de Campos Finitos existen muchos problemas importantes tales como: dado un primo p y un entero n encontrar un polinomio irreducible de grado n sobre el campo finito base $\mathbb{F}_p$ con el fin de construir un campo finito $\mathbb{F}_q$, con q elementos donde $q = p^n$.

Teniendo el campo finito $\mathbb{F}_q$ en general es complicado hallar un elemento g del campo tal que $< g >= \mathbb{F}_q^* = \mathbb{F}_q - \{0\}$, es decir, un generador del grupo cíclico. Otro problema es el logaritmo discreto sobre campos finitos. Este problema consiste en que si conocemos $g, \alpha \in \mathbb{F}_q$ tal que $\alpha = g^x$, entonces es computacionalmente imposible encontrar x.

Además de contener en si los problemas anteriores, los campos finitos tienen muchas aplicaciones, en Combinatoria, Teoría de Códigos, Criptografía y otras áreas de las Matemáticas e Ingeniería [MuSh, Sh92, MeOoVa96]. En Criptografía, por ejemplo, el criptosistema de llave privada que desplazará al DES, el Rijndael [DaRi98] consiste de transformaciones realizadas sobre $\mathbb{F}_{2^8}$, donde, la aritmética de este campo juega un papel muy importante. Por otro lado, en la criptografía de llave pública, los criptosistemas se basan en problemas computacionalmente difíciles de resolver como lo es la descomposición prima de un número grande, usado en RSA [RiShAd78]. El problema del logaritmo discreto usado en el criptosistema de ElGamal [El85] y en el intercambio de llaves Diffie-Hellman [DiHe76]. La versión aditiva del problema del logaritmo discreto (es decir, encontrar x tal que $\alpha = xg$ conociendo α y g) utilizada en el criptosistema basado en curvas elípticas sobre campos finitos [Ko87, Mi86].

En Teoría de Códigos, uno de los principales problemas es corregir los errores que se producen al enviar mensajes a través de un canal "ruidoso".

Estos mensajes consisten de elementos de un alfabeto finito, o simplemente de 0's y 1's, así generalmente son vistos como números binarios, y a su vez como elementos de un campo finito. Los métodos usados para corregir errores están fuertemente ligados a propiedades de los campos finitos [MaSl77, Pl98].

En combinatoria, por ejemplo, los campos finitos se usan para construir cuadrados latinos. Un cuadrado latino de orden n es un arreglo $n \times n$, sobre n símbolos distintos con la propiedad de que cada renglón y cada columna contenga cada uno de los n símbolos exactamente una vez. Por otro lado, los cuadrados latinos son útiles en el diseño de experimentos estadísticos. Más aplicaciones a la combinatoria se pueden encontrar en [MuSh, Ha67, Ry63, WaStWa72].

Con el avance del poder computacional, los campos finitos utilizados son cada vez más grandes, creciendo también la dificultad de realizar operaciones como suma, multiplicación y exponenciación, es decir, aritmética sobre $\mathbb{F}_{q^n}$. Por ejemplo, en el criptosistemas de curvas elípticas se hacen sumas, multiplicación y división para el cálculo y suma de puntos racionales sobre un campo finito, por ejemplo con 2^{100} elementos. Así muchos investigadores alrededor del mundo se han dado a la tarea de encontrar algoritmos eficientes para realizar esta aritmética.

Para efectuar estas operaciones primero debemos representar a los elementos de $\mathbb{F}_{q^n}$ de una forma especial. Como $\mathbb{F}_{q^n}$ es un espacio vectorial sobre $\mathbb{F}_q$, existe una base

$$\{w_1, w_2, \ldots, w_n\}$$

tal que todo elemento de $\mathbb{F}_{q^n}$ lo podemos escribir como

$$\gamma = \sum_{i=1}^{n} a_i w_i$$

con $a_i \in \mathbb{F}_q$.

Entonces el problema radica en que al realizar las operaciones mencionadas, debemos obtener el resultado en términos de la base escogida. Por ejemplo, cuando multiplicamos

$$\gamma_1 = \sum_{i=1}^{n} a_i w_i \quad \text{y} \quad \gamma_2 = \sum_{i=1}^{n} b_i w_i \quad con\ a_i, b_i \in \mathbb{F}_q$$

$$\gamma_1 \cdot \gamma_2 = (\sum_{i=1}^{n} a_i w_i)(\sum_{i=1}^{n} b_i w_i) = \sum_{i,j=1}^{n} c_{ij} w_i w_j$$

debemos calcular los productos $w_i w_j$ en términos de la base

$$\{w_1, w_2, \ldots, w_n\}.$$

La eficiencia de la aritmética dependerá de la base utilizada. Un tipo de bases utilizadas son las *bases polinomiales* las cuales solo consideran hasta la $n-1$ potencia de un elemento $\alpha \in \mathbb{F}_{q^n}$, es decir,

$$\{1, \alpha, \alpha^2, \ldots, \alpha^{n-1}\}.$$

Este tipo de base es la más utilizada en implementaciones de algunos criptosistemas sobre campos de característica 2, como es el caso del Rijndael.

Otro tipo de base son las llamadas *bases normales.* Si los $n-1$ conjugados de un elemento $\alpha \in \mathbb{F}_{q^n}$ y α son linealmente independientes se obtiene una base normal, es decir,

$$N = \{\alpha, \alpha^q, \ldots, \alpha^{q^{n-1}}\}.$$

Una de las ventajas de utilizar este tipo de base es que al elevar un elemento representado en la base N, a una potencia de q, esta operación sólo equivale a un corrimiento, cuyo costo computacional es insignificante.

El elemento del campo finito,

$$\gamma = \sum_{i=0}^{n-1} a_i \alpha^i$$

se puede representar por el vector $(a_0, a_1, \ldots, a_{n-1})$, entonces

$$\gamma^q = (\sum_{i=0}^{n-1} a_i \alpha^i)^q = (a_0, a_1, \ldots, a_{n-1})^q = (a_1, \ldots, a_{n-1}, a_0).$$

Un campo finito $\mathbb{F}_{q^n}$ siempre contiene al menos una base normal. El problema ahora es construir o buscar bases normales, y de estas elegir las

que realicen las operaciones aritméticas eficientemente, es decir, con el menor costo computacional.

Un ejemplo son las llamadas *bases normales óptimas*. Estas bases se caracterizan porque la representación única de los productos de sus elementos, tienen el menor número posible de coeficientes distintos de cero, lo cual facilita las operaciones. Hay criterios que nos ayudan a determinar cuando un campo finito $\mathbb{F}_{q^n}$ tiene una base normal óptima.

Otra construcción que será considerada dentro de este trabajo fue dada a conocer en 1995 por S. Gao, J. von zur Gathen y D. Panario [GaGaPa95]. Esta construcción retoma los períodos de Gauss, utilizando una adaptación de éstos para campos finitos, los cuales generarán una base normal bajo ciertas condiciones.

El objetivo de este trabajo es presentar algunos métodos para efectuar aritmética rápida sobre un campo finito. Desarrollando este tema en 4 capítulos, en el primero veremos nociones básicas acerca de los campos finitos. En el capítulo 2 se tratarán las bases polinomiales y su aritmética. El tercer capítulo trata acerca de las bases normales, incluyendo como se construyen las bases normales óptimas, la construcción y aritmética de las bases normales generadas con los períodos de Gauss, y en el último capítulo veremos algunos criptosistemas sobre campos finitos, cuya implementación eficiente requiere que cualquier operación entre sus elementos sea lo mas rápida posible (en el menor tiempo posible). Se incluye un apéndice acerca de complejidad computacional.

Capítulo 1

Campos finitos

Los orígenes de la Teoría de Campos Finitos se remontan a los siglos $XVII$ y $XVIII$, con eminentes matemáticos como Pierre de Fermat (1601-1665), Leonhard Euler (1707-1783), Joseph Louis Lagrange (1736-1813) y Adrien Marie Legendre (1752-1833), que trabajaron los llamados *campos finitos primos.* Se puede decir que los más importantes fueron los trabajos de Carl Friedrich Gauss (1777-1855) y Evariste Galois (1811-1832).

El estudio de los campos finitos ha tenido un gran desarrollo en los últimos 60 años, debido principalmente a problemas surgidos de sus aplicaciones. Este tema siempre era una pequeña parte de libros de Algebra Moderna hasta que, en 1983 se edita el primer libro totalmente dedicado a campos finitos [LiNi83].

1.1 Definición y propiedades básicas

Un campo finito, es un anillo conmutativo $(K, +, *)$ tal que $K - \{0\}$ es un grupo conmutativo bajo la operación " $*$ ".

Ejemplo 1.1.1. *Los*

$$\mathbb{Z}/p\mathbb{Z} = \mathbb{Z}_p = \{\overline{0}, \overline{1}, \overline{2}, \overline{3}, \ldots, \overline{p-1}\}$$

donde p es primo son campos finitos.

Definición 1.1.2. *Para un primo p, sean $\mathbb{F}_p = \{0, 1, 2, \ldots, p-1\}$ y*

$$\psi : \mathbb{Z}/p\mathbb{Z} \longmapsto \mathbb{F}_p, \quad \psi(\overline{a}) = a \quad \textit{para } a = 0, 1, 2 \ldots, p-1.$$

Entonces $\mathbb{F}_p$ obtiene la estructura de campo finito a través del mapeo ψ, y es llamado Campo de Galois de orden p.

Una manera natural de construir un campo finito, por ejemplo $\mathbb{F}_{p^m}$ es la siguiente: primero debemos encontrar algún polinomio $f(x) \in \mathbb{F}_p[x]$ de grado m irreducible sobre $\mathbb{F}_p$ (llamado *campo base*). Sea el anillo cociente,

$$\frac{\mathbb{F}_p[x]}{< f(x) >} = \{a_0 + a_1 x + \cdots + a_{m-1}x^{m-1} + < f(x) > : a_i \in \mathbb{F}_p\}.$$

Si consideramos $\alpha \equiv x \bmod f(x)$, entonces $f(\alpha) = 0$, como $< f(x) >$ es un ideal máximo el cociente es un campo. Con esto el campo finito $\mathbb{F}_{p^m}$, lo identificamos con el conjunto,

$$\mathbb{F}_{p^m} = \{a_0 + a_1\alpha + \cdots + a_{m-1}\alpha^{m-1} : a_i \in \mathbb{F}_p \text{ y la relación } f(\alpha) = 0\}$$

donde las operaciones de suma y multiplicación estan determinadas por el polinomio $f(x)$ utilizado. Aseguramos que siempre existe un campo finito por el Teorema 2.5 [LiNi83] dice que para todo primo p y todo entero n existe un campo finito con p^n elementos. Cualquier campo finito con $q = p^n$ elementos es isomorfo al campo de descomposición del polinomio $x^q - x$ sobre $\mathbb{F}_p$.

Podemos usar el campo $\mathbb{F}_{p^m}$ para construir otro campo finito, usando ahora $\mathbb{F}_{p^m}$ como campo base para obtener el campo finito $\mathbb{F}_{q^n}$ donde $q = p^m$.

Ejemplo 1.1.3. *Consideremos como campo base a los binarios $\mathbb{F}_2 = \{0, 1\}$, construiremos el campo $\mathbb{F}_{2^3}$ usando el polinomio $f(x) = x^3 + x + 1$.*

El campo finito es:

$$\mathbb{F}_{2^3} = \{a_0 + a_1\alpha + a_2\alpha^2 : a_i \in \mathbb{F}_2 \text{ y } \alpha^3 + \alpha + 1 = 0\}$$

explícitamente este conjunto es

$$\mathbb{F}_8 = \{0, 1, \alpha, 1 + \alpha, \alpha^2, 1 + \alpha^2, \alpha + \alpha^2, 1 + \alpha + \alpha^2\}.$$

Sea $\mathbb{F}_{q^n}$ el campo finito definido por un polinomio irreducible $f(x)$ de grado n. $\mathbb{F}_{q^n}$ es isomorfo como espacio vectorial a $\{\mathbb{F}_q\}^n$.

A través de

$$\theta : \mathbb{F}_{q^n} \longmapsto \{\mathbb{F}_q\}^n,\ \theta(a_0 + a_1\alpha + a_2\alpha^2 + \cdots + a_{n-1}\alpha^{n-1}) = (a_0, a_1, a_2, a_3, \ldots, a_{n-1}).$$

+	(000)	(001)	(010)	(011)	(100)	(101)	(110)	(111)
(000)	(000)	(001)	(010)	(011)	(100)	(101)	(110)	(111)
(001)	(001)	(000)	(011)	(010)	(101)	(100)	(111)	(110)
(010)	(010)	(011)	(000)	(001)	(110)	(111)	(100)	(101)
(011)	(011)	(010)	(001)	(000)	(111)	(110)	(101)	(100)
(100)	(100)	(101)	(110)	(111)	(000)	(001)	(010)	(011)
(101)	(101)	(100)	(111)	(110)	(001)	(000)	(011)	(010)
(110)	(110)	(111)	(100)	(101)	(010)	(011)	(000)	(001)
(111)	(111)	(110)	(101)	(100)	(011)	(010)	(001)	(000)

Tabla 1.1: Suma de los elementos de $\mathbb{F}_8$

*	(000)	(001)	(010)	(011)	(100)	(101)	(110)	(111)
(000)	(000)	(000)	(000)	(000)	(000)	(000)	(000)	(000)
(001)	(000)	(001)	(010)	(011)	(100)	(101)	(110)	(111)
(010)	(000)	(010)	(100)	(110)	(011)	(001)	(111)	(101)
(011)	(000)	(011)	(110)	(101)	(111)	(100)	(001)	(010)
(100)	(000)	(100)	(011)	(111)	(110)	(010)	(101)	(001)
(101)	(000)	(101)	(001)	(100)	(010)	(111)	(011)	(110)
(110)	(000)	(110)	(111)	(001)	(101)	(011)	(010)	(100)
(111)	(000)	(111)	(101)	(010)	(001)	(110)	(100)	(011)

Tabla 1.2: Multiplicación de los elementos de $\mathbb{F}_8$

donde $\alpha \equiv \mod f(x)$.

La representación vectorial es muy útil para almacenar y simplificar el manejo de los elementos de un campo finito. Las operaciones de suma y multiplicación de $\mathbb{F}_{2^3}$ (usando vectores en los cuales omitimos las comas) están definidas por las tablas 1.1 y 1.2.

El conjunto de elementos $a \in \mathbb{F}_{p^n}$ distintos de cero, denotado por $\mathbb{F}_{p^n}^*$ es un grupo multiplicativo .

Teorema 1.1.4. $\mathbb{F}_q^*$ *es un grupo cíclico.*

Demostración. Supongamos que $\mathbb{F}_q$ tiene más de tres elementos. Sea $h =$

$p_1^{r_1}p_2^{r_2}\cdots p_m^{r_m}$ la descomposición en factores primos de $h = q-1$. Para toda i, $1 \leq i \leq m$, el polinomio $x^{\frac{h}{p_i}} - 1$ tiene a lo más $\frac{h}{p_i}$ soluciones en $\mathbb{F}_q$. Ya que $\frac{h}{p_i} < h$, se sigue que hay elementos distintos de cero en $\mathbb{F}_q$ que no son raíces de este polinomio. Sea a_i uno de esos elementos y $b_i = a_i^{h/p_i^{r_i}}$. Tenemos $b_i^{p_i^{r_i}} = 1$, así el orden de b_i es un divisor de $p_i^{r_i}$ y por lo tanto de la forma $p_s^{r_s}$ con $0 \leq s_i \leq r_i$.

También b_i cumple

$$b_i^{p_i^{r_i-1}} = a_i^{h/p_i} \neq 1,$$

entonces el orden de b_i es $p_i^{r_i}$. Afirmamos que el elemento $b = b_1 b_2 \cdots b_m$ tiene orden h. Supongamos lo contrario, que el orden de b es un divisor de h y divisor de al menos uno de los m enteros h/p_i, $1 \leq i \leq m$, digamos h/p_1. Entonces tenemos

$$1 = b^{h/p_1} = b_1^{h/p_1} b_2^{h/p_1} \cdots b_m^{h/p_1}.$$

Si $2 \leq i \leq m$, entonces $p_i^{r_i}$ divide a h/p_1, y así $b_i^{h/p_1} = 1$, por lo tanto $b_1^{h/p_1} = 1$, esto implica que el orden de b_1 debe dividir a h/p_1, lo cual es imposible ya que el orden de b_1 es $p_i^{r_i}$. Así, $\mathbb{F}_q^*$ es un grupo cíclico y tiene como generador a b.

□

Observación 1.1.5. *Para cualquier $\beta \neq 0 \in \mathbb{F}_q^*$,*

$$\beta^s = 1 \quad \text{si y sólo si} \quad ord(\beta)|s.$$

Lema 1.1.6. *Sea $\alpha \in \mathbb{F}_q^*$. Si el $ord(\alpha) = t$, entonces $ord(\alpha^i) = \frac{t}{(i,t)}$ para $1 \leq i \leq q-2$.*

Demostración. Sea $d = (i,t)$, entonces $\alpha^{i(t/d)} = \alpha^{t(i/d)} = (\alpha^t)^{(i/d)} = 1$, por la observación 1.1.5 se cumple que $\text{ord}(\alpha^i)|(t/d)$. Sea $s = \text{ord}(\alpha^i)$, es decir, $\alpha^{is} = 1$ y nuevamente por la observación 1.1.5 obtenemos que $t|is$. Como $d = (i,t)$, entonces existen enteros a y b tales que $ia + tb = d$. Multiplicando esta ecuación por s, obtenemos $isa + tsb = sd$, pero como $t|is$, se sigue que $t|ds$, así se cumple $(t/d)|s$, por lo tanto $(t/d)|\text{ord}(\alpha^i)$, con lo que $\text{ord}(\alpha^i) = t/d$.

□

Teorema 1.1.7. *Sea $\mathbb{F}_q$ un campo con q elementos y t un entero positivo. Si $t|(q-1)$ entontes $\mathbb{F}_q^*$ tiene $\phi(t)$ elementos de orden t, donde ϕ es la función ϕ de Euler.*

Demostración. Primero veamos que hay $\phi(t)$ elementos de orden t, si existe $\alpha \in \mathbb{F}_q^*$ tal que $\text{ord}(\alpha) = t$. Entonces estos elementos estan en el conjunto $\{1, \alpha, \alpha^2, \ldots, \alpha^{t-1}\}$, pero por el lema 1.1.6 las potencias α^i que tienen orden t son aquellas que $(i,t) = 1$, que son precisamente $\phi(t)$ las que cumplen esta condición. Ahora denotemos por $\psi(t)$ al número de elementos de orden t en $\mathbb{F}_q^*$, esto para cada divisor t de $q-1$. Como todo elemento debe tener un orden que divide a $q-1$,

$$\sum_{t|q-1} \psi(t) = q - 1. \tag{1.1}$$

esta relación también la cumple la función ϕ de Euler ([NiZu85] Teorema 2.17, pag. 45), combinando ambas relaciones obtenemos

$$\sum_{t|q-1} (\phi(t) - \psi(t)) = 0$$

pero como hay $\phi(t)$ elementos de orden t, entonces $\phi(t) - \psi(t) \geq 0$, para toda t, así $\phi(t) = \psi(t)$ para toda $t|q-1$.

□

Lema 1.1.8. *Si g es un generador de $\mathbb{F}_q^*$ entonces g^j también es un generador si y sólo si $(j, q-1) = 1$. Por lo tanto $\mathbb{F}_q^*$ tiene $\phi(q-1)$ generadores diferentes.*

Demostración. Se sigue directamente del teorema 1.1.7.

□

Definición 1.1.9 (Elemento primitivo). *Un elemento $\alpha \in \mathbb{F}_q$ de orden $q-1$, es decir, un generador de $\mathbb{F}_q^*$, es llamado elemento primitivo del campo $\mathbb{F}_q$.*

Ejemplo 1.1.10. *Todos los elementos distintos de 1 en $\mathbb{F}_{2^3}^*$ (ver ejemplo 1.1.3) son primitivos.*

Consideremos las potencias de α,

$$\begin{aligned}
\alpha^0 &= 1 \\
\alpha^1 &= \alpha \\
\alpha^2 &= \alpha^2 \\
\alpha^3 &= \alpha + 1 \\
\alpha^4 &= \alpha^2 + \alpha \\
\alpha^5 &= \alpha^2 + \alpha + 1 \\
\alpha^6 &= \alpha^2 + 1
\end{aligned}$$

De acuerdo con el lema 1.1.8 tenemos tantos generadores como

$$\phi(8-1) = \phi(7) = 6,$$

es decir, $\mathbb{F}_{2^3}$ tiene 6 generadores.

1.2 Polinomios mínimos

Sea $\mathbb{F}_q$ un campo finito, con $q = p^m$ elementos donde p es primo y m un entero positivo. Sabemos que $\mathbb{F}_q$ puede ser visto como un espacio vectorial sobre $\mathbb{F}_p$, de dimensión m. Sea $\alpha \in \mathbb{F}_q$ y consideremos las $m+1$ potencias de α,

$$1, \alpha, \alpha^2, \ldots, \alpha^m \quad \in \mathbb{F}_q.$$

Como $\mathbb{F}_q$ tiene dimensión m, las $m+1$ potencias de α deben ser linealmente dependientes sobre $\mathbb{F}_p$, es decir, deben existir elementos $c_i \in \mathbb{F}_q$, $0 \leq i \leq m$, no todos cero tales que

$$c_0 + c_1\alpha + \cdots + c_m\alpha^m = 0,$$

por lo que α es raíz del polinomio

$$c(x) = c_0 + c_1 x + \cdots + c_m x^m. \tag{1.2}$$

El elemento α también puede ser raíz de otros polinomios, consideremos el conjunto formado por estos, es decir,

$$S(\alpha) = \{f(x) \in \mathbb{F}_p[x] : f(\alpha) = 0\}.$$

El conjunto $S(\alpha)$ tiene al menos un elemento de grado $\leq m$. Sea $p(x)$ el polinomio de menor grado distinto de cero en $S(\alpha)$ y sea $f(x)$ cualquier otro

polinomio en $S(\alpha)$. Por el algoritmo de la división para polinomios, existen polinomios $q(x)$, $r(x)$ tales que

$$f(x) = q(x)p(x) + r(x), \ \ \text{grad}(r) < \text{grad}(r).$$

Pero como $f(\alpha) = p(\alpha) = 0$, obtenemos que también $r(\alpha) = 0$, entonces $r(x)$ es un polinomio de menor grado que tiene como raíz a α lo cual no puede ser si $p(x)$ es el mínimo, por lo tanto $r(x) \equiv 0$, por lo que se cumple:

$$p(x)|f(x), \quad \forall f(x) \in S(\alpha). \tag{1.3}$$

Además $S(\alpha)$ es un ideal y $p(x)$ es un generador de éste.

Definición 1.2.1 (Polinomio mínimo). *$p(x) \in \mathbb{F}_q[x]$ es el polinomio mínimo de $\alpha \in \mathbb{F}_q$ si es mónico y satisface la condición 1.3.*

Teorema 1.2.2. *Sea $\mathbb{F}_q$ un campo finito con $q = p^m$ elementos. Asociado con cada $\alpha \in \mathbb{F}_q$, existe un único polinomio $p(x) \in \mathbb{F}_p[x]$ con las siguientes propiedades:*

(a) $p(\alpha) = 0$

(b) $grad(p) \leq m$

(c) Si $f(x)$ es otro polinomio en $\mathbb{F}_p[x]$ con $f(\alpha) = 0$, entonces

$$p(x)|f(x)$$

Demostración. Se desprende de la definición 1.2.1.

□

Ejemplo 1.2.3. *Considerar el campo finito del ejemplo 1.1.3, los polinomios mínimos de cada elemento son:*

elemento	*polinomio mínimo*
0	x
1	$x+1$
α	x^3+x+1
α^2	x^3+x+1
α^3	x^3+x^2+1
α^4	x^3+x+1
α^5	x^3+x^2+1
α^6	x^3+x^2+1

Lema 1.2.4. *Sea el campo $K = \mathbb{F}_{q^n}$ y su subcampo $\mathbb{F} = \mathbb{F}_q$. Entonces un elemento $\beta \in K$ está en el subcampo $\mathbb{F}$ si y sólo si $\beta^q = \beta$.*

Demostración. $\Longrightarrow$) Sea $\beta \in \mathbb{F}$. Por el Teorema de Lagrange tenemos $\beta^{q-1} = 1$, multiplicando β obtenemos $\beta^q = \beta$.
$\Longleftarrow$) Si $\beta^q = \beta$ por lo que $\beta^{q-1} = 1$, es decir, el orden de β divide al orden de $\mathbb{F}$, entonces, $\beta \in \mathbb{F}$.

□

Observación 1.2.5. *Sean $\alpha_1, \alpha_2, \ldots, \alpha_t$ elementos de $\mathbb{F}_{p^m}$ campo finito de característica p. Entonces*

$$(\alpha_1 + \alpha_2 + \cdots + \alpha_t)^{p^k} = (\alpha_1)^{p^k} + (\alpha_2)^{p^k} + \cdots + (\alpha_t)^{p^k}$$

para $k = 1, 2, 3, \ldots$

Definición 1.2.6 (Elementos conjugados). *Los conjugados de $\alpha \in \mathbb{F}_{p^m}$ son los elementos:*

$$\alpha, \alpha^p, \alpha^{p^2}, \alpha^{p^3}, \cdots, \alpha^{p^{m-1}}$$

(recordando que $\alpha^{p^m} = \alpha$).

Sea $\alpha \in \mathbb{F}_p$ todos los α^{p^i} son raices del polinomio mínimo de α. Entonces, los conjugados de $\alpha \in \mathbb{F}_{p^m}$ son todos distintos si y sólo si el polinomio mínino de α sobre $\mathbb{F}_p$ tiene grado m.

Si el grado del polinomio mínimo es d, un divisor de m, entonces los conjugados de α distintos son $\alpha, \alpha^p, \alpha^{p^2}, \ldots, \alpha^{p^{d-1}}$, cada uno se repite m/d veces.

Una forma sencilla de calcular el número de conjugados distintos, es encontrando el menor entero d tal que $q^d \equiv 1 \bmod (\mathrm{ord}(\alpha))$, puesto que se cumple $\alpha^{q^d} = \alpha$.

Ejemplo 1.2.7. *Usando el ejemplo 1.1.3, los conjugados de α son α^2 y $\alpha^{2^2} = \alpha^4$, y los conjugados de α^3 son $(\alpha^3)^2 = \alpha^6$ y $(\alpha^3)^{2^2} = \alpha^5$ (recordemos que $\alpha^7 = 1$). Podemos observar del ejemplo 1.2.3 que los elementos que son conjugados entre ellos tienen el mismo polinomio mínimo.*

Teorema 1.2.8. *Sea $\mathbb{F}_{p^m}$ un campo finito. Si $\alpha \in \mathbb{F}_{p^m}$, entonces el polinomio mínimo de α sobre $\mathbb{F}_p$ es:*

$$f_\alpha(x) = \prod_{i=0}^{d-1} (x - \alpha^{p^i})$$

donde d es el número de conjugados distintos de α.

Demostración. Supongamos que $\alpha, \alpha^p, \ldots, \alpha^{p^{d-1}}$ son los conjugados distintos de α. Estos son raices del polinomio mínimo. Entonces el polinomio $f(x) = (x-\alpha)(x-\alpha^p)\cdots(x-\alpha^{p^{d-1}})$ divide al polinomio mínimo de α.

Desarrollamos $f(x)$ en potencias de x, esto es:

$$(x-\alpha)(x-\alpha^p)\cdots(x-\alpha^{p^{d-1}}) = A_d x^d + A_{d-1}x^{d-1} + \cdots + A_1 x + A_0.$$

Si elevamos este polinomio a la potencia p obtenemos:

$$f(x)^p = (x^p-\alpha^p)(x^p-\alpha^{p^2})\cdots(x^p-\alpha^{p^d}) = A_d^p x^{pd} + A_{d-1}^p x^{p(d-1)}$$
$$+\cdots+ A_1^p x^p + A_0^p = f(x^p),$$

es decir,

$$A_d x^{pd} + A_{d-1}x^{p(d-1)} + \cdots + A_1 x^p + A_0 = A_d^p x^{pd} + A_{d-1}^p x^{p(d-1)} + \cdots + A_1^p x^p + A_0^p$$

Por lo tanto los coeficientes del polinomio $A_i = A_i^p$ para $0 \leq i \leq d$ y así pertenecen a $\mathbb{F}_p$. Así $f(x)$ tiene menor grado que el polinomio mínimo y lo divide por lo tanto concluimos que $f(x)$ es el polinomio mínimo de α.

□

1.3 Trazas y Normas

Las funciones *traza* y *norma* son herramientas muy importantes en particular en la teoría de campos finitos.

Definición 1.3.1 (Traza). *Sea $K = \mathbb{F}_{q^n}$ un campo con q^n elementos. La traza de $\alpha \in K$ sobre el campo $\mathbb{F} = \mathbb{F}_q$ es la función $Tr : \mathbb{F}_{q^n} \mapsto \mathbb{F}_q$ definida como la suma de sus conjugados, es decir,*

$$Tr_{\mathbf{F}}^K(\alpha) = \alpha + \alpha^q + \alpha^{q^2} + \cdots + \alpha^{q^{n-1}}.$$

Definición 1.3.2 (Norma). *Con las mismas hipótesis de la definición anterior, la norma del elemento α es la función $N : \mathbb{F}_{q^n} \mapsto \mathbb{F}_q$:*

$$N_{\mathbf{F}}^K(\alpha) = \alpha \cdot \alpha^q \cdots \alpha^{q^{n-1}}.$$

Ahora veremos algunas propiedades de la traza y la norma.

Teorema 1.3.3. *Para toda $\alpha, \beta \in \mathbb{F}_{q^n}$ tenemos*

(a) $Tr(\alpha) \in \mathbb{F}_q$	*(a')* $N(\alpha) \in \mathbb{F}_q$
(b) $Tr(\alpha+\beta) = Tr(\alpha) + Tr(\beta)$	*(b')* $N(\alpha\beta) = N(\alpha)N(\beta)$
(c) $Tr(\lambda\alpha) = \lambda Tr(\alpha)$ si $\lambda \in \mathbb{F}_q$	*(c')* $N(\lambda\alpha) = \lambda^n N(\alpha)$ si $\lambda \in \mathbb{F}_q$
(d) $Tr(\alpha^q) = Tr(\alpha)$	*(d')* $N(\alpha^q) = N(\alpha)$
(e) Tr es un mapeo suprayectivo	*(e')* N es un mapeo suprayectivo

Demostración. Ver Teorema 2.23 y 2.28 [LiNi83]. □

Ejemplo 1.3.4. *Consideremos el campo $\mathbb{F}_{3^3}$, definido con el polinomio irreducible $f(x) = x^3 + 2x + 1$, es decir,*

$$\mathbb{F}_{3^3} = \{a_0 + a_1\alpha + a_2\alpha^2 : a_0, a_1, a_2 \in \mathbb{F}_3 \text{ y } \alpha^3 + 2\alpha + 1 = 0\}$$

las trazas de los elementos α y α^2 sobre $\mathbb{F}_3$ son:

$$Tr(\alpha) = \alpha + \alpha^3 + \alpha^{3^2} = 0,$$

$$Tr(\alpha^2) = \alpha^2 + (\alpha^2)^3 + (\alpha^2)^{3^2} = 2.$$

Ejemplo 1.3.5. *Usemos el mismo campo del ejemplo anterior, y calculemos las normas de los elementos α y α^2:*

$$N(\alpha) = \alpha \cdot \alpha^3 \cdot \alpha^{3^2} = 2,$$

$$N(\alpha^2) = \alpha^2 \cdot (\alpha^2)^3 \cdot (\alpha^2)^{3^2} = 1.$$

Teorema 1.3.6. *Sean $\mathbb{F}_{q^n}$ y $\mathbb{F} = \mathbb{F}_q$ campos finitos (consideremos ambos como espacios vectoriales sobre $\mathbb{F}_q$). Entonces las transformaciones lineales de $\mathbb{F}_{q^n}$ en $\mathbb{F}_q$ son exactamente los mapeos L_β, $\beta \in \mathbb{F}_{q^n}$, donde $L_\beta(\alpha) = Tr(\beta\alpha)$ para toda $\alpha \in \mathbb{F}_{q^n}$. Además, si $\beta \neq \gamma$ $L_\beta \neq L_\gamma$.*

Demostración. Por el teorema 1.3.3 (b) y (c) L_β es una tranformación lineal de K en $\mathbb{F}_q$. Para $\beta, \gamma \in \mathbb{F}_{q^n}$ distintos tenemos

$$L_\beta(\alpha) - L_\gamma(\alpha) = Tr(\beta\alpha) - Tr(\gamma\alpha) = Tr((\beta - \gamma)\alpha) \neq 0$$

para alguna $\alpha \in \mathbb{F}_{q^n}$, entonces $L_\beta \neq L_\gamma$. Como $\mathbb{F}_{q^n}$ tiene q^n elementos distintos también tenemos q^n tranformaciones lineales L_β distintas. Por otro lado, toda transformación lineal de $\mathbb{F}_{q^n}$ en $\mathbb{F}_q$ puede ser obtenida asignando elementos arbitrarios de $\mathbb{F}_q$ a una base dada de $\mathbb{F}_{q^n}$ sobre $\mathbb{F}_q$. Puesto que hay q^n maneras posibles de hacer esta elección. Por lo tanto L_β corresponde a alguna transformación lineal.

□

Capítulo 2

Bases polinomiales

La aritmética sobre un campo finito $\mathbb{F}_{q^n}$, depende de la elección de la base. Puesto que las propiedades de la base elegida influyen en la eficiencia de los algoritmos de multiplicación y exponenciación (la complejidad de la suma es la misma sin importar la base). Las bases de un campo finito (como espacio vectorial) se dividen en dos tipos principalmente, las *bases polinomiales* que veremos en este capítulo, y las *bases normales* que se verán en el siguiente. Primero veremos algunas consideraciones sobre bases.

2.1 Bases

Si el conjunto $\{\alpha_0, \alpha_1, \ldots, \alpha_{n-1}\}$ es una base de $\mathbb{F}_{q^n}$ sobre $\mathbb{F}_q$, entonces cada elemento γ de $\mathbb{F}_{q^n}$ puede ser representado de forma única como:

$$\gamma = a_0\alpha_0 + a_1\alpha_1 + \cdots + a_{n-1}\alpha_{n-1} \text{ con } a_i \in \mathbb{F}_q \text{ para } 0 \leq i \leq n-1.$$

Definición 2.1.1 (Base dual). *Sean* $\mathbf{B}_1 = \{\alpha_0, \alpha_1, \ldots, \alpha_{n-1}\}$ *y* $\mathbf{B}_2 = \{\beta_0, \beta_1, \ldots, \beta_{n-1}\}$ *bases de* $\mathbb{F}_{q^n}$ *sobre* $\mathbb{F}_q$. *Decimos que* $\mathbf{B}_1$ *y* $\mathbf{B}_2$ *son duales si* $Tr(\alpha_i \cdot \beta_j) = \delta_{ij}$ *(delta de Kronecker), es decir,*

$$\delta_{ij} = \begin{cases} 1, & \text{si } i = j \\ 0, & \text{si } i \neq j \end{cases}$$

El siguiente resultado asegura la existencia y unicidad de la base dual.

Teorema 2.1.2. *Para cualquier base* $\mathbf{B}_1$ *de* $\mathbb{F}_{q^n}$ *sobre* $\mathbb{F}_q$ *existe una única base dual.*

Demostración. Para cualquier $\gamma \in \mathbb{F}_{q^n}$ sea

$$\gamma = \sum_{i=0}^{n-1} \gamma_i \alpha_i \qquad \text{donde } \gamma_i \in \mathbb{F}_q$$

la única representación de γ en términos de la base $\mathbf{B_1}$. Como Tr es una transformación lineal de $\mathbb{F}_{q^n}$ sobre $\mathbb{F}_q$, por el teorema 1.3.6 existen únicos $\beta_i \in \mathbb{F}_{q^n}$ tales que

$$\gamma_i = Tr(\beta_i \gamma), \quad 0 \leq i \leq n-1.$$

Entonces

$$\gamma = \sum_{i=0}^{n-1} Tr(\beta_i \gamma)\alpha_i$$

para cualquier $\gamma \in \mathbb{F}_{q^n}$. En particular, para $\gamma = \alpha_j$, se tiene

$$\alpha_j = \sum_{i=0}^{n-1} Tr(\beta_i \alpha_j)\alpha_i,$$

lo cual implica que $Tr(\alpha_i \cdot \beta_j) = \delta_{ij}$.

Además, si consideramos una combinación lineal del conjunto $\mathbf{B_2} = \{\beta_0, \beta_1, \ldots, \beta_{n-1}\}$, $\sum_i d_i \beta_i = 0$, $d_i \in \mathbb{F}_q$, entonces $(\sum_i d_i \beta_i)\alpha_j = 0$ y como

$$Tr(\sum_i d_i \beta_i \alpha_j) = 0 = \sum_i d_i Tr(\beta_i \alpha_j) = 0 + 0 + \cdots + d_j + \cdots + 0$$

lo cual implica que $d_j = 0$, para $j = 0, 1, 2 \ldots, n-1$. De esta forma $\mathbf{B_2}$ es la única base dual a B_1.

□

Tenemos la siguiente caracterización para una base.

Teorema 2.1.3. *El conjunto de elementos* $\mathbf{B_1} = \{\alpha_0, \alpha_1, \ldots \alpha_{n-1}\}$ *es una base de* $\mathbb{F}_{q^n}$ *sobre* $\mathbb{F}_q$ *si y sólo si la matriz* A *es no singular donde*

$$A = \begin{bmatrix} \alpha_0 & \alpha_1 & \cdots & \alpha_{n-1} \\ \alpha_0^q & \alpha_1^q & \cdots & \alpha_{n-1}^q \\ \vdots & \vdots & \vdots & \vdots \\ \alpha_0^{q^{n-1}} & \alpha_1^{q^{n-1}} & \cdots & \alpha_{n-1}^{q^{n-1}} \end{bmatrix}$$

Demostración. Si $\mathbf{B}_1$ es una base, por el teorema anterior existe la base dual $\mathbf{B}_2 = \{\beta_0, \beta_1, \ldots, \beta_{n-1}\}$ y sea

$$B = \begin{bmatrix} \beta_0 & \beta_1 & \cdots & \beta_{n-1} \\ \beta_0^q & \beta_1^q & \cdots & \beta_{n-1}^q \\ \vdots & \vdots & \vdots & \vdots \\ \beta_0^{q^{n-1}} & \beta_1^{q^{n-1}} & \cdots & \beta_{n-1}^{q^{n-1}} \end{bmatrix}$$

entonces $BA = I_n$, la $n \times n$ matriz identidad, y A es no singular.

Recíprocamente supongamos que A es no singular. Si

$$\sum_{i=0}^{n-1} c_i \alpha_i = 0,\ c_i \in \mathbb{F}_q,$$

elevando ambos lados a la potencia q^j, obtenemos $\sum_{i=0}^{n-1} c_i \alpha_i^{q^i} = 0$, y se tiene que $A\bar{c} = 0$ (donde $\bar{c} = (c_0, c_1, \ldots, c_{n-1})$). Como A es no singular, $\bar{c} = 0$ y $\mathbf{B}_1$ es una base.

□

Denotando por $Tr(\mathbf{B}_1^T \mathbf{B}_1)$ a la matriz cuyas entradas ij son los productos $\alpha_i \alpha_j$ del resultado anterior se tiene el siguiente corolario.

Corolario 2.1.4. $\mathbf{B}_1 = \{\alpha_0, \alpha_1, \ldots, \alpha_{n-1}\}$ *es una base de* $\mathbb{F}_{q^n}$ *sobre* $\mathbb{F}_q$ *si y sólo si* $Tr(\mathbf{B}_1^T \mathbf{B}_1)$ *es no singular.*

Demostración. Si A está definida como en el teorema 2.1.3, entonces $Tr(\mathbf{B}_1^T \mathbf{B}_1) = A^T A$, la cual es no singular si y sólo si A es no singular. El resultado se sigue del teorema 2.1.3.

□

2.2 Aritmética

Definición 2.2.1 (Base polinomial). *Sea* $\beta \in \mathbb{F}_{q^n}$, *si*

$$BP = \{1, \beta, \beta^2, \ldots, \beta^{n-1}\}$$

son linealmente independientes, decimos que BP es una base polinomial de $\mathbb{F}_{q^n}$ *sobre* $\mathbb{F}_q$.

Recordando la construcción de un campo finito descita en la sección 1.1, y considerando α tal que $f(\alpha) = 0$ (donde f es irreducible), en la definición del campo

$$\mathbb{F}_{q^n} = \{\sum_{i=0}^{n-1} a_i\alpha^i : a_i \in \mathbb{F}_q \quad \text{y } f(\alpha) = 0\},$$

claramente el conjunto

$$BP = \{1, \alpha, \alpha^2, \ldots, \alpha^{n-1}\},$$

es una base polinomial de $\mathbb{F}_{q^n}$ sobre $\mathbb{F}_q$.

Ejemplo 2.2.2. *Sea $\mathbb{F}_{3^6}$ el campo finito definido por el polinomio $f(x) = x^6 + 2x^5 + 2x + 2$. Una base polinomial de $\mathbb{F}_{3^6}$ es:*

$$BP_{\mathbb{F}_{3^6}} = \{1, \alpha, \alpha^2, \alpha^3, \alpha^4, \alpha^5\},$$

donde $\alpha^6 = \alpha^5 + \alpha + 1$.

Las operaciones que se requieren realizar sobre $\mathbb{F}_{q^n}$ son: suma, multiplicación y exponenciación[1].

Para efectuar estas operaciones, primero escogemos una base polinomial

$$BP_{\mathbb{F}_{q^n}} = \{1, \alpha, \alpha^2, \ldots, \alpha^{n-1}\},$$

después debemos dar algoritmos para obtener los resultados en términos de la base $BP_{\mathbb{F}_{q^n}}$.

2.2.1 Suma

Sean $\gamma_1, \gamma_2 \in \mathbb{F}_{q^n}$, cuyas representaciones (únicas) en términos de la base $BP_{\mathbb{F}_{q^n}}$ son:

$$\gamma_1 = \sum_{i=0}^{n-1} a_i\alpha^i, \quad \gamma_2 = \sum_{i=0}^{n-1} b_i\alpha^i \quad \text{con} \quad a_i, b_i \in \mathbb{F}_q. \tag{2.1}$$

[1] Calcular división se reduce a multiplicación y exponenciación, puesto que $\mathbb{F}_{q^n}^*$ es cíclico

Consideremos los elementos γ_1 y γ_2 cuyas representaciones están dadas por (2.1). La suma de estos elementos es la adición usual (vectorial), es decir,

$$\gamma_1 + \gamma_2 = \sum_{i=0}^{n-1} a_i\alpha^i + \sum_{i=0}^{n-1} b_i\alpha^i = \sum_{i=0}^{n-1}(a_i + b_i)\alpha^i.$$

Donde claramente $(a_i + b_i) \in \mathbb{F}_q$.

Ejemplo 2.2.3. *Sean* $\alpha^5+2\alpha^2+2\alpha+1$, $\alpha^5+2\alpha^4+\alpha^3+2 \in \mathbb{F}_{3^6}$. *Entonces su suma es:*

$$\begin{aligned}(\alpha^5 + 2\alpha^2 + 2\alpha + 1) + (\alpha^5 + 2\alpha^4 + \alpha^3 + 2) =&(1 + 1)\alpha^5 + (0 + 2)\alpha^4 + (0 + 1)\alpha^3 \\ &+ (2 + 0)\alpha^2 + (2 + 0)\alpha + (1 + 2) \\ =&2\alpha^5 + 2\alpha^4 + \alpha^3 + 2\alpha^2 + 2\alpha\end{aligned}$$

Si lo representamos en términos de sus coordenadas obtenemos:

$$(100221) + (121002) = (221220).$$

La complejidad de la operación suma es $O(n)$, ya que lo que realizamos es a lo más n operaciones (ver Apéndice A). Además esta complejidad es la misma si usamos cualquier base.

2.2.2 Multiplicación

Sean γ_1 y γ_2 como en (2.1). Ahora veremos la multiplicación de estos dos elementos representada en términos de la base polinomial $BP_{\mathbb{F}_{q^n}}$.

Directamente esto es:

$$\gamma_1 \cdot \gamma_2 = (\sum_{i=0}^{n-1} a_i\alpha^i)(\sum_{i=0}^{n-1} b_i\alpha^i) = \sum_{k=0}^{2n-2} c_k\alpha^k$$

$$\text{donde} \quad c_k = \sum_{k=i+j} a_i b_j.$$

Como podemos observar, necesitamos expresar las potencias α^k para $n \leq k \leq 2n-2$, en términos de la base $BP_{\mathbb{F}_{q^n}}$, para finalmente realizar sumas. Notemos que esto es muy similar a la multiplicación de polinomios, si precalculamos las potencias necesarias este algoritmo tiene complejidad

$O(n^2)$, ya que se realizan $2n-2$ sumas de c_k que a su vez se lleva a lo más n multiplicaciónes, siendo $O(2n^2-2n)=O(n^2)$.

Efectuar esta multiplicación de forma más eficiente consistirá en mejorar los algoritmos existentes de mulplicación entre polinomios. Explicaremos un algoritmo para multiplicar polinomios el cual utiliza la Transformación Discreta de Fourier (TDF)[GeCzLa, CaKa91], la complejidad de este algoritmo es $O(nlogn)$. Para explicar en que consiste esta transformación, primero necesitamos desarrollar algunos conceptos.

Transformación Discreta de Fourier

Sean $\{x_0,\ldots,x_{n-1}\}$ tales que $x_i \in \mathbb{F}_q$ para $0 \leq i \leq n-1$, un conjunto de n puntos, deseamos calcular una transformación $T : \mathbb{F}_q^n \longmapsto \mathbb{F}_q^n$:

$$T_{(x_0,\ldots,x_{n-1})}(a_0,\ldots,a_{n-1}) = (\hat{a}_0,\ldots,\hat{a}_{n-1}) \tag{2.2}$$

$$\text{donde} \quad \hat{a}_i = a_0 + a_1 x_i + \cdots + a_{n-1} x_i^{n-1}.$$

Si $a(x) = \sum_{i=0}^{n-1} a_i x^i$, la transformación (2.2) equivale a evaluar $a(x)$ en los puntos x_i.

El costo o complejidad de evaluar polinomios utilizando el método de Horner es $O(n)$ (ver [Kn81] 467-469). Por lo que si lo usamos para evaluar la transformación (2.2) tendremos complejidad $O(n^2)$. Esta complejidad es reducida utilizando el siguiente procedimiento.

Primero consideremos el problema de la evaluación de $a(x)$ con grado n par, y escribimos a $a(x)$ de la forma:

$$a(x) = b(x^2) + xc(x^2) \tag{2.3}$$

$$\text{donde} \quad b(y) = a_0 + a_2 y + \cdots + a_{n-2} y^{n/2-1}$$

$$\text{y} \quad c(y) = a_1 + a_3 y + \cdots + a_{n-1} y^{n/2-1}.$$

Ejemplo 2.2.4. *El polinomio* $a(x) = 2x^6 - 7x^4 + 3x^3 + x^2 - 12x + 5$ *se puede escribir en términos de dos polinomios de grado menor o igual a 3 como sigue:*

$a(x) = b(x^2) + xc(x^2)$
$b(y) = 5 + y - 7y^2 + 2y^3$
$c(y) = -12 + 3y.$

Lema 2.2.5. *Sean* $x_i \in \mathbb{F}_q$ *para* $0 \leq i \leq n-1$ *un conjunto de n puntos, con n par, que satisfacen la siguiente condición de simetría:*

$$x_{n/2+i} = -x_i \tag{2.4}$$

$$para \quad 0 \leq i \leq n/2 - 1$$

Si $M(n)$ *es el costo de evaluar un polinomio de grado* $n-1$ *en n puntos, entonces* $M(1) = 0$, *y* $M(n) = 2M(\frac{n}{2}) + s(\frac{n}{2})$ *para alguna constante s.*

Demostración. Como

$$x_0^2 = x_{n/2}^2,\ x_1^2 = x_{n/2+1}^2,\ \ldots,\ x_{n/2-1}^2 = x_{n-1}^2,$$

entonces sólo hay $\frac{n}{2}$ cuadrados distintos.

Un polinomio de grado $\leq n-1$ puede ser evaluado usando los polinomios $b(y)$ y $c(y)$ (definidos en 2.3) en los $\frac{n}{2}$ puntos

$$\{x_0^2, \ldots, x_{n/2-1}^2\}$$

así con $a(x) = b(x^2) + xc(x^2)$ obtenemos la evaluación que queremos. Notemos que si $n = 1$, $M(n)$ es el costo de evaluar un polinomio constante, lo cual no requiere ninguna operación por lo tanto M(1)=0.

En el caso cuando $n > 1$ tenemos que evaluar 2 polinomios de grado $\frac{n}{2}$ lo cual contribuye con $M(\frac{n}{2})$, además realizar $\frac{n}{2}$ multiplicaciones para obtener los cuadrados de x_i, llevará digamos s sumas y restas, es decir,

$$M(n) = 2M(\frac{n}{2}) + s(\frac{n}{2})$$

□

En el procedimiento de multiplicación se usará el lema 2.2.5 en forma recursiva. Para esto necesitamos que se cumpla la simetría. Veremos que esto en efecto pasa con ciertos elementos de un campo finito.

Definición 2.2.6 (Elementos de **Fourier**). *Cuando ω es una raíz n-ésima primitiva de la unidad en un campo finito $\mathbb{F}_q$, el conjunto de n elementos*

$$\{1, \omega, \omega^2, \ldots, \omega^{n-1}\}$$

son llamados elementos o puntos de Fourier.

La transformación (2.2), usando los elementos de Fourier es llamada la *Transformación Discreta de Fourier* TDF. Esta es:

$$TDF(a_0, \ldots, a_{n-1}) = T_{(1,\omega,\ldots,\omega^{n-1})}(a_0, \ldots, a_{n-1}).$$

Una vez que hemos seleccionado los ω_i, utilizaremos la matriz asociada a esta transformación usando la base canónica

$$\{(1, 0, \ldots, 0), (0, 1, \ldots, 0), \ldots, (0, 0, \ldots, 1)\}$$

del espacio vectorial $\mathbb{F}^n$. A esta matriz la denotaremos como $[TDF]$.

Ejemplo 2.2.7. *Sea $\mathbb{F}_{37}$ el campo de los enteros módulo 37. Construiremos la matriz $[TDF]$ de la transformación $TDF : \mathbb{F}_{37}^4 \mapsto \mathbb{F}_{37}^4$*

Primero debemos encontrar una raíz cuarta primitiva de la unidad en $\mathbb{F}_{37}$. Iniciamos calculando las potencias de 2:

$$< 2 >= \{2, 4, 8, 16, 32, 27, 17, 34, 31, 25, 13, 26, 15, \cdots, 1\},$$

por lo tanto 2 es un generador de $\mathbb{F}_{37}$.

Usando el lema 1.1.6, $2^9 = 31$ tiene orden 4, así los elementos de Fourier son

$$\{1, 31, 31^2, 31^3\} = \{1, 31, 36, 6\}$$

La $[TDF]$ asociada a la transformación $T_{(1,31,36,6)}$ usando la base canónica $\{(1,0,0,0), (0,1,0,0), (0,0,1,0), (0,0,0,1)\}$ es:

$$[T_{(1,31,36,6)}] = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 31 & 36 & 6 \\ 1 & 36 & 1 & 36 \\ 1 & 6 & 36 & 31 \end{bmatrix}$$

Lema 2.2.8. *Si $\omega \in \mathbb{F}_q$ es una raíz n-ésima primitiva de la unidad, entonces los n elementos de Fourier satisfacen la condición de simetría del lema 2.2.8.*

Demostración. Notando que

$$((\omega)^{n/2+j})^2 = \omega^{n+2j} = \omega^n \omega^{2j} = \omega^{j^2}.$$

Se tiene que

$$(\omega^{n/2+j})^2 - (\omega^j)^2 = 0$$

de aquí

$$(\omega^{n/2+j} + \omega^j)(\omega^{n/2+j} - \omega^j) = 0$$

Si $\omega^{n/2+j} = \omega^j$ entonces $\omega^{n/2} = 1$, lo cual contradice que ω es una raíz primitiva y por lo tanto:

$$\omega^{n/2+j} = -\omega^j$$

□

Lema 2.2.9. *Sea n un entero par y ω una n-ésima raíz primitiva de la unidad de un campo finito $\mathbb{F}_q$, entonces*

(a) ω^2 es una raíz $\frac{n}{2}$-ésima de la unidad,

(b) los $\frac{n}{2}$ cuadrados satisfacen la condición de simetría del lema 2.2.8.

Demostración. (a) Que ω^2 es una raíz de la unidad se sigue del lema 1.1.6, ya que ese es precisamente su orden,
(b) Se desprende del lema 2.2.8.

□

Teorema 2.2.10. *Sea $\omega \in \mathbb{F}_q$ una raíz n-ésima primitiva de la unidad. Entonces, la transformación TDF definida a partir de los n elementos de Fourier puede ser calculada en $O(nlogn)$ operaciones.*

Demostración. Probaremos el teorema cuando $n = 2^m$ para algún entero m. El lema 2.2.9 implica que el costo de evaluar un polinomio de grado $n-1$ es la función $M(n)$, la cual satisface la recursión:

$$M(1) = 0$$

$$M(2^k) = 2 \cdot M(2^{k-1}) + s \cdot 2^{k-1} \quad \text{para } k \geq 1.$$

El costo se simplifica a

$$\begin{aligned} M(n) = M(2^m) =& 2 \cdot M(2^{m-1}) + s \cdot 2^{m-1} \\ =& 2(2M(2^{m-2}) + s \cdot 2^{m-2}) + s \cdot 2^{m-1} \\ =& 2^2 M(2^{m-2}) + 2 \cdot s \cdot 2^{m-1} \\ =& 2^3 M(2^{m-3}) + s2^{m-1} \cdot 3 = \cdots = 2^m M(1) + s2^{m-1} \cdot m \\ =& s \cdot 2^{m-1} \cdot m = s \cdot \frac{n}{2} logn \end{aligned}$$

con lo que obtenemos que la complejidad es $O(nlogn)$.

□

La transformación inversa de Fourier.

Ahora queremos encontrar si existe, para un conjunto de puntos

$$\{x_0, \ldots, x_{n-1}\} \subset \mathbb{F}$$

la transformación $(T_{(x_0,x_1,\ldots,x_{n-1})})^{-1}$.

Definición 2.2.11. *La inversa de la transformación discreta de Fourier*[2] *para un conjunto de elementos de Fourier del campo finito $\mathbb{F}_q$ está definida por la transformación $ITDF : \mathbb{F}^n \longmapsto \mathbb{F}^n$*

$$ITDF_{(1,\omega,\ldots,\omega^{n-1})}(q_0, \ldots, q_{n-1}) = (\bar{q}_0, \ldots, \bar{q}_{n-1}) \tag{2.5}$$

donde $\bar{q}_j = n^{-1} \sum_{k=0}^{n-1} q_k \cdot (\omega^{-j})^k$, y ω es una raíz n-ésima primitiva de la unidad en el campo $\mathbb{F}_q$ (notemos que si $\mathbb{F}_q$ tiene una raíz n-ésima primitiva de la unidad, entonces, $n|q-1$ y por lo tanto existe $n^{-1} mod\ car(\mathbb{F}_q)$).

Teorema 2.2.12. *Las transformaciones TDF e $ITDF$ son inversas.*

Demostración.

$$T_{(1,\omega,\ldots,\omega^{n-1})}(a_0, \ldots, a_{n-1}) = (\hat{a}_0, \ldots, \hat{a}_{n-1})$$

como en la ecuación (2.2), esto es, cada

$$\hat{a}_i = \sum_{j=0}^{n-1} a_j (\omega^i)^j \quad para\ i = 0, 1, \ldots, n-1$$

[2]En este caso si existe, puesto que los puntos de Fourier son linealmente independientes por ser raices de la unidad.

Sustituyendo los $\hat{a}_i$ en $ITDF$ obtenemos:

$$ITDF_{(1,\omega,\ldots,\omega^{n-1})}(\hat{a}_0,\ldots,\hat{a}_{n-1}) = n^{-1}(\sum_{i=0}^{n-1}\hat{a}_i,\ldots,\sum_{i=0}^{n-1}\hat{a}_i(\omega^{-(n-1)})^i)$$

Para cualquier entero $0 \leq k \leq n-1$ tenemos

$$\begin{aligned} n^{-1}\cdot\sum_{i=0}^{n-1}\hat{a}_i(\omega^{-k})^i =& n^{-1}\sum_{i=0}^{n-1}\left(\sum_{j=0}^{n-1}a_j\omega^{ij}\right)\cdot\omega^{-ki} \\ =& n^{-1}\sum_{j=0}^{n-1}a_j\left(\sum_{i=0}^{n-1}\omega^{(j-k)i}\right) \end{aligned}$$

Como ω es raíz n-ésima del polinomio

$$x^n - 1 = (x-1)(x^{n-1}+x^{n-2}+\cdots+x+1),$$

ω^s con $0 < s < n$ también es raíz de $x^n - 1 = 0$. Puesto que $(\omega^s)^n = (\omega^n)^s = 1$ y $\omega^s \neq 1$, se cumple que ω^s es raíz del polinomio $x^{n-1}+x^{n-2}+\cdots+x+1 = 0$. Sustituyendo ω^s en el polinomio $\sum_{i=0}^{n-1}x^i = 0$ obtenemos $\sum_{i=0}^{n-1}(\omega^s)^i = 0$, si $0 < s < n$, por lo que tenemos los casos:

$$\sum_{i=0}^{n-1}(\omega^{(j-k)i}) = \begin{cases} 0, & \text{si } 0 < j-k < n \\ n, & \text{si } j-k = 0 \text{ o } j-k = n. \end{cases}$$

Entonces $n^{-1}\sum_{j=0}^{n-1}a_j(\sum_{i=0}^{n-1}\omega^{(j-k)i}) = n^{-1}a_k\sum_{i=0}^{n-1}\omega^0 = n^{-1}a_k n = a_k$ si $j-k=0$. El caso $j-k=n$ no sucede ya que $j-k<n$ y $0 \leq j,k < n$. De esta forma obtenemos:

$$ITDF_{(1,\omega,\ldots,\omega^{n-1})}(\hat{a}_0,\ldots,\hat{a}_{n-1}) = (a_0,\ldots,a_{n-1}),$$

y por lo tanto $ITDF$ es la inversa de TDF.

□

Ejemplo 2.2.13. *Calculemos sobre el campo finito $\mathbb{F}_{37}$ la inversa $ITDF$ para TDF definida con $\omega = 31$ (ver ejemplo 2.2.7).*

Primero calculamos las potencias negativas de ω:

$$(1,\omega^{-1},\omega^{-2},\omega^{-3}) = (1,6,36,31).$$

Sea $[ITDF]$ la matriz asociada a la transformación $ITDF_{(1,31,36,6)}$ usando la base canónica de $\mathbb{F}_{37}^4$. Entonces

$$[ITDF] = 4^{-1}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 6 & 36 & 31 \\ 1 & 36 & 1 & 36 \\ 1 & 31 & 36 & 6 \end{bmatrix} = \begin{bmatrix} 28 & 28 & 28 & 28 \\ 28 & 20 & 9 & 17 \\ 28 & 9 & 28 & 9 \\ 28 & 17 & 9 & 20 \end{bmatrix}$$

Se puede checar que la matriz $[TDF]$ del ejemplo 2.2.7 es inversa de $[ITDF]$ módulo 37.

Notemos que

$$TDF(a_0, \ldots, a_{n-1}) = T_{(1,\omega,\ldots,\omega^{n-1})} \text{ y}$$

$$ITDF(a_0, \ldots, a_{n-1}) = n^{-1} T_{(1,\omega^{-1},\ldots,\omega^{-(n-1)})}(a_0, \ldots, a_{n-1}).$$

Por lo tanto las transformaciones TDF y $ITDF$ tienen la misma complejidad, la cual es $O(nlogn)$ por el teorema 2.2.10.

Regresando ahora a lo que nos interesa que es la multiplicación en un campo finito, podemos ver a cada elemento de $\mathbb{F}_{q^n}$, como un polinomio en $\mathbb{F}_q[\alpha]$ de grado $n-1$.

Algoritmo 2.2.14. *Para multiplicar 2 elementos definidos como en la ecuación (2.1), vistos como polinomios en $\mathbb{F}_q[\alpha]$ de grado $n-1$.*

1. *Encontrar una potencia de 2 más grande o igual que $2n$. Sea $N = 2^m$ la menor potencia que es mayor a $2n$.*

2. *Encontrar una raíz N-ésima primitiva de la unidad en $\mathbb{F}_q$, en caso de que no tenga avanzamos a la siguiente extensión digamos $\mathbb{F}_{q^i}$ donde $i|n$, que contenga una raíz N-ésima* [3].

3. *Calculamos $[TDF]$ y $[ITDF]$ de la raíz N-ésima.*

[3] En caso de que tengamos que usar una extensión tan grande como el campo en el que queremos trabajar, este método no mejorará la complejidad de la multiplicación.

4. *Sean* $\overline{\gamma_1}$ *y* $\overline{\gamma_2}$ *los vectores formados por los coeficientes de* $\gamma_1(\alpha)$ *y* $\gamma_2(\alpha)$, *y tantos ceros hasta obtener un vector con* N *entradas. Por ejemplo si* $\gamma_1(\alpha) = \sum_{i=0}^{n-1} a_i\alpha^i$, *entonces:*

$$\overline{\gamma_1} = (a_0, \ldots, a_{n-1}, 0, 0, \ldots, 0).$$

5. *Multiplicar la matriz* $[TDF]$ *a cada uno de los vectores* $\overline{\gamma_1}$ *y* $\overline{\gamma_2}$ *obteniendo:*

$$A = [TDF](\overline{\gamma_1}) = (A_0, \ldots, A_{N-1})$$
$$B = [TDF](\overline{\gamma_2}) = (B_0, \ldots, B_{N-1}).$$

6. *Calcular*

$$C_i = A_iB_i \quad \textit{donde } i = 0, \ldots, N-1$$
$$C = (C_1, \cdots, C_{N-1})$$

7. *Calcular*

$$\overline{\gamma_1 \cdot \gamma_2} = \overline{M} = [ITDF] \cdot C.$$

Así

$$\gamma_1(\alpha) \cdot \gamma(\alpha) = \sum_{i=0}^{N-1} M_i\alpha^i.$$

Con este procedimiento lo que tendríamos que calcular previamente son las potencias α^j para $n \leq j < N$.

Ejemplo 2.2.15. *Calculemos la multiplicación sobre el campo finito* $\mathbb{F}_{3^4}$ *definido por el polinomio* $f(x) = x^4 + x + 2$. *Consideremos* $\beta^4 + \beta + 2 = 0$ *y así una base polinomial de* $\mathbb{F}_{3^4}$ *es:*

$$BP = \{1, \beta, \beta^2, \beta^3\}.$$

Vamos a multiplicar polinomios sobre $\mathbb{F}_3[\beta]$ de grado 3. En este caso como se tiene grado 3 $n = 4$, $N = 2^m \geq 2(4)$, por lo que $N = 8$ y $m = 3$. Claramente $\mathbb{F}_3$ no tiene una raíz octava primitiva de la unidad. Entonces consideramos el campo finito $\mathbb{F}_{3^2}$ definido por el polinomio $g(x) = x^2 + 1$

$$\mathbb{F}_{3^2} = \{0, 1, 2, \alpha, \alpha + 1, \alpha + 2, 2\alpha, 2\alpha + 1, 2\alpha + 2\}.$$

con $\alpha^2 = 2$, cuyo generador $\alpha + 1$ es precisamente una raíz octava de la unidad.

Ahora debemos calcular $[TDF]$ y $[ITDF]$ asociadas a $\omega = \alpha + 1$:

$$[TDF] = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1+\alpha & 2\alpha & 1+2\alpha & 2 & 2+2\alpha & \alpha & 2+\alpha \\ 1 & 2\alpha & 2 & \alpha & 1 & 2\alpha & 2 & \alpha \\ 1 & 1+2\alpha & \alpha & 1+\alpha & 2 & 2+\alpha & 2\alpha & 2+2\alpha \\ 1 & 2 & 1 & 2 & 1 & 2 & 1 & 2 \\ 1 & 2+2\alpha & 2\alpha & 2+\alpha & 2 & 1+\alpha & \alpha & 1+2\alpha \\ 1 & \alpha & 2 & 2\alpha & 1 & \alpha & 2 & 2\alpha \\ 1 & 2+\alpha & \alpha & 2+2\alpha & 2 & 1+2\alpha & 2\alpha & 1+\alpha \end{bmatrix}$$

$$[ITDF] = \begin{bmatrix} 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 2 & 1+2\alpha & 2\alpha & 1+\alpha & 1 & 2+\alpha & \alpha & 2+2\alpha \\ 2 & 2\alpha & 1 & \alpha & 2 & 2\alpha & 1 & \alpha \\ 2 & 1+\alpha & \alpha & 1+2\alpha & 1 & 2+2\alpha & 2\alpha & 2+\alpha \\ 2 & 1 & 2 & 1 & 2 & 1 & 2 & 1 \\ 2 & 2+\alpha & 2\alpha & 2+2\alpha & 1 & 1+2\alpha & \alpha & 1+\alpha \\ 2 & \alpha & 1 & 2\alpha & 2 & \alpha & 1 & 2\alpha \\ 2 & 2+2\alpha & \alpha & 2+\alpha & 1 & 1+\alpha & 2\alpha & 1+2\alpha \end{bmatrix}$$

Estamos listos para calcular las potencias de β $\{\beta^4, \beta^5, \beta^6, \beta^7\}$ expresadas en términos de la base $BP = \{1, \beta, \beta^2, \beta^3\}$.

Usando que $\beta^4 + \beta + 2 = 0$:

$\beta^4 = 2\beta + 1$,

$\beta^5 = \beta\beta^4 = 2\beta^2 + \beta$ y

$\beta^6 = \beta\beta^5 = 2\beta^3 + \beta^2$.

Calculemos β^7 por medio del algoritmo 2.2.14. Entonces los dos polinomios a multiplicar serán $a(\beta) = \beta^3$ y $b(\beta) = \beta^4 = 2\beta + 1$.

Con estos polinomios construimos los vectores

$$\overline{a} = (0, 0, 0, 1, 0, 0, 0, 0)$$

$$\overline{b} = (1, 2, 0, 0, 0, 0, 0, 0).$$

Calculamos

$$A = TDF\overline{a}^T = (1, 1 + 2\alpha, \alpha, 1 + \alpha, 2, 2 + \alpha, 2\alpha, 2 + 2\alpha)$$
$$B = TDF\overline{b}^T = (0, 2\alpha, 1 + \alpha, \alpha, 2, 2 + \alpha, 1 + 2\alpha, 2 + 2\alpha).$$

Después,

$$C = (0, 2 + 2\alpha, 2 + \alpha, 2 + \alpha, 1, \alpha, 2 + 2\alpha, 2\alpha).$$

Aplicando la matriz $[ITDF]$ a C obtenemos:

$$M = (0, 0, 0, 1, 2, 0, 0, 0),$$

este vector como polinomio en $\mathbb{F}_3[\beta]$ es $\beta^3 + 2\beta^4$, entonces $\beta^7 = \beta^3 + 2\beta^4$. Finalmente como $\beta^4 = 2\beta + 1$ resulta

$$\beta^7 = 2 + \beta + \beta^3.$$

En general la mulplicación de dos elementos $a = \sum_{i=0}^{3} a_i\beta^i, b\sum_{i=0}^{3} b_i\beta^i \in \mathbb{F}_3^4$, siendo los vectores de longitud 8 formados por sus coeficientes $\overline{a} = (a_0, a_1, a_2, a_3, 0, 0, 0, 0)$ y $\overline{b} = (b_0, b_1, b_2, b_3, 0, 0, 0, 0)$.

Al aplicar a los vectores $\overline{a}$ y $\overline{b}$ el procedimiento definido por el algoritmo 2.2.14 resulta

$$M = (m_0, m_1, m_2, m_3, m_4, m_5, m_6, m_7).$$

Finalmente como ya hemos calculado las potencias β^4, β^5, β^6 y β^7 obtenemos:

$$\begin{aligned} a \cdot b =& m_0 + m_1\beta + m_2\beta^2 + m_3\beta^3 + m_4\beta^4 + m_5\beta^5 + m_6\beta^6 + m_7\beta^7 \\ =& m_0 + m_4 + m_7 + (m_1 + 2m_4, m_5 + 2m_7)\beta + (m_2 + 2m_5 + m_6)\beta^2 \\ &+ (m_3 + 2m_6 + m_7)\beta^3. \end{aligned}$$

En el caso de que se utilice otro polinomio sólo se calculan las potencias β^4, β^5, β^6 y β^7 en términos del nuevo polinomio. Puesto que servirán las mismas transformaciones TDF y $ITDF$.

La complejidad del algoritmo 2.2.14 para multiplicar dos polinomios de grado $n - 1$, usando TDF se calcula así primero notemos que se realizan 3 evaluaciones de TDF, y $N \geq 2n$, entonces la complejidad final de la multiplicación es $O(2(n)\log 2(n)) = O(n\log n)$. Mejorando así el costo en comparación de realizar $O(n^2)$ operaciones [GeCzLa, Sc77, CaKa91].

2.2.3 Exponenciación

La exponenciación es realizar la multiplicación del mismo elemento varias veces, así que mejorar este proceso tiene que ver con el tratamiento que hagamos del exponente a calcular. Esto es, el objetivo de una exponenciación eficiente es efectuar el menor número de multiplicaciones. Veremos algunos métodos para economizar la cantidad de operaciones.

El método binario

Sea $\alpha \in \mathbb{F}_{q^n}$. Para calcular $(\alpha)^m$ usando el método binario, primero se debe considerar la representación binaria del exponente m

$$m = \sum_{i=0}^{l} m_i 2^i,$$

donde $l = \log_2(m)$. Definimos $\nu(m)$ (llamado el *peso de Hamming*) como el número de $m_i \neq 0$.

Como $\alpha^m = (((\alpha^{2m_l}\alpha^{m_{l-1}})^2 \ldots)^2\alpha^{m_1})^2\alpha^{m_0}$. Se realizan $\nu(m)$ operaciones, si se han calculado previamente las potencias α^{2^i} para $0 \leq i \leq l$.

Ejemplo 2.2.16. *Calculando α^{21} donde $\alpha \in \mathbb{F}_{3^2}$.*

Como $21 = 1 + 0 \cdot 2 + 1 \cdot 2^2 + 0 \cdot 2^3 + 1 \cdot 2^4$, así

$$\alpha^{21} = (((\alpha^2\alpha^0)^2\alpha)^2\alpha^0)^2\alpha = \alpha^{16}\alpha^4\alpha,$$

la cual requiere sólo tres operaciones.

Método r-ario

De manera similar al binario para un entero $r \geq 2$, en este caso se considera la expansión r-aria de m

$$m = \sum_{i=0}^{s} m_i r^i,$$

donde $s = \log_r(m)$. Además se calculan las potencias α^{r^i} para $0 \leq i \leq s$. En este caso el número máximo de operaciones es $\nu(m)r$.

Método de Factorización

En este caso suponemos que es posible representar a m de la forma $m = kl$, donde k es el menor divisor primo de m y $l > 1$ o en otro caso $m = kl + 1$. Para calcular α^m primero calculamos $X_1 = \alpha^k$ y $X_2 = \alpha^l$ o en el otro caso $X_2 = \alpha^l \cdot \alpha$. Después calculamos simplemente $\alpha^m = X_1 X_2$.

Este método es mejor que el binario, pero factorizar un entero m grande, es muy difícil.

Cadenas de adición

El problema de multiplicar potencias de α se puede reducir a encontrar los enteros cuya suma sea el exponente que se desea calcular. Este método es útil cuando se quiere calcular un exponente fijo.

Definición 2.2.17 (Cadena de adición). *Una cadena de adición para un entero m es una sucesión de enteros*

$$1 = a_0,\ a_1,\ \ldots,\quad a_r = m$$

que cumple la propiedad

$$a_i = a_j + a_k,\quad \textit{para algunas } k \leq j < i,\ \textit{para toda } i = 1, 2, \ldots, r.$$

Ejemplo 2.2.18. *Calculemos algunas cadenas de adición para $m = 21$.*

Claramente las siguientes son cadenas de adición de 21:

$$a_0 = 1,\ a_1 = 2,\ a_3 = 3,\ a_4 = 5,\ a_5 = 7,\ a_6 = 14,\ a_7 = 21$$

$$a_0 = 1,\ a_1 = 2,\ a_3 = 3,\ a_4 = 5,\ a_5 = 6,\ a_6 = 10,\ a_7 = 11,\ a_8 = 21.$$

Notemos que en cualquiera de los dos casos para calcular α^{21} sólo se multiplican 2 potencias. Esto es

$$\alpha^{21} = \alpha^7 \alpha^{14} = \alpha^{10} \alpha^{11}.$$

Podemos ver que se hacen más cálculos previos en la segunda cadena de adición. A diferencia con el método binario, sólo se multiplican dos potencias de α en vez de tres.

El menor subíndice r de una cadena de adición se denota por $l(m)$. El propósito ahora es encontrar la cadena de adición más corta. Algunas referencias con respecto a cadenas de adición son [BrGoMcWi92, Ro94, Sc75, Ya90]. Y en general sobre exponenciación [Kn81].

Capítulo 3

Bases Normales

En este capítulo estudiaremos las bases llamadas *normales*. Cuando un elemento de un campo finito $\mathbb{F}_{q^n}$ es representado usando este tipo de base, elevar este elemento a la potencia q equivale a un corrimiento cuyo costo computacional es insignificante. Veremos dos tipos de bases normales: las óptimas y las generadas por los períodos de Gauss. Se siguió de cerca los siguientes trabajos: [MuOnVa89, GaGaPa95, GaGaPa98, GaVa95]. Primero veremos algunos conceptos básicos sobre bases normales.

3.1 Bases Normales

Definición 3.1.1 (Base Normal). *Sea $\gamma \in \mathbb{F}_{q^n}$. Donde $\mathbb{F}_{q^n}$ una extensión de $\mathbb{F}_q$ ($q = p^m$ con p primo). Si los conjugados de γ*

$$B = \{\gamma, \gamma^q, \ldots, \gamma^{q^{n-1}}\},$$

son linealmente independientes se dice que B es una base normal de $\mathbb{F}_{q^n}$ sobre $\mathbb{F}_q$. Además γ es llamado elemento normal .

Ejemplo 3.1.2. *Utilizando nuevamente el ejemplo 1.1.3, una base normal de*

$$\mathbb{F}_8 = \{0, 1, \alpha, \alpha+1, \alpha^2, \alpha^2+1, \alpha^2+\alpha, \alpha^2+\alpha+1\}$$

sobre $\mathbb{F}_2$ es:

$$\{(\alpha+1), (\alpha+1)^2, (\alpha)^{2^2}\} = \{\alpha+1, \alpha^2+1, \alpha^2+\alpha+1\},$$

donde $\alpha^3 = \alpha + 1$.

En [LiNi83], Teorema 2.35 página 60, se demuestra, que siempre existe una base normal de $\mathbb{F}_{q^n}$ sobre $\mathbb{F}_q$.

Sea $BN = \{\alpha, \alpha^q, \ldots, \alpha^{q^{n-1}}\}$ una base normal de $\mathbb{F}_{q^n}$ sobre $\mathbb{F}_q$, sean los elementos $\gamma_1, \gamma_2 \in \mathbb{F}_{q^n}$:

$$\gamma_1 = \sum_{i=0}^{n-1} a_i \alpha^{q^i} \quad \text{y} \quad \gamma_2 = \sum_{i=0}^{n-1} b_i \alpha^{q^i}, \quad a_i, b_i \in \mathbb{F}_q.$$

La suma de γ_1 y γ_2 es como se vió en la sección 2.2.1, es decir, se suman los coeficientes.

Antes de ver la multiplicación consideremos el siguiente lema.

Lema 3.1.3. *Sea $\overline{\gamma_1} = (a_0, a_1, \ldots, a_{n-1})$ el vector formado por los coeficientes de $\gamma_1 = \sum_{i=0}^{n-1} a_i \alpha^{q^i}$. Si elevamos γ_1 a una potencia q^j con $j \leq n$, entonces el vector que representa la potencia es un corrimiento de j lugares a la derecha de $\overline{\gamma_1}$, es decir,*

$$\overline{\gamma_1^{q^j}} = (a_{n-j}, a_{n-j+1}, \ldots, a_{n-1}, a_0, \ldots, a_{n-j-2}, a_{n-j-1}).$$

Demostración. Calculando

$$\begin{aligned}
(\gamma_1)^{q^j} &= (\sum_{i=0}^{n-1} a_i \alpha^{q^i})^{q^j} \\
&= \sum_{i=0}^{n-1} (a_i \alpha^{q^i})^{q^j} \quad \text{por la observación 1.2.5} \\
&= \sum_{i=0}^{n-1} a_i \alpha^{q^{i+j}} \quad \text{por el lema 1.2.4} \\
&= \sum_{i=0}^{n-1-j} a_i \alpha^{q^{i+j}} + \sum_{i=n-j}^{n-1} a_i \alpha^{q^{i+j}} \\
&= \sum_{i=j}^{n-j-1+j} a_{i-j} \alpha^{q^i} + \sum_{i=0}^{j-1} a_{i+(n-j)} \alpha^{q^{i+n}} \\
&= \sum_{i=0}^{j-1} a_{i+(n-j)} \alpha^{q^i} + \sum_{i=j}^{n-1} a_{i-j} \alpha^{q^i} \quad \text{por el lema 1.2.4}
\end{aligned}$$

entonces $\overline{(\gamma_1)^{q^j}} = (a_{n-j}, a_{n-j+1}, \ldots, a_{n-1}, a_0, \ldots, a_{n-j-2}, a_{n-j-1})$

□

Ejemplo 3.1.4. *Consideremos el campo* $\mathbb{F}_{3^3}$ *definido por el polinomio irreducible* $f(x) = x^3 + x^2 + x + 2$, *con* $\alpha \equiv x \bmod f(x)$

$$BN = \{\alpha, \alpha^3, \alpha^{3^2}\} = \{\alpha, 2\alpha^2 + 2\alpha + 1, \alpha^2 + 1\}.$$

Sea $\beta \in \mathbb{F}_{3^3}$ *cuya representación es* $2\alpha + \alpha^3 + \alpha^{3^2}$ *considerando los coeficientes* $\overline{\beta} = (2, 1, 1)$, *calculemos*

$$\beta^3 = (2\alpha + \alpha^3 + \alpha^{3^2})^3 = 2\alpha^3 + \alpha^{3^2} + \alpha^{3^3} = \alpha + 2\alpha^3 + \alpha^{3^2}$$

que podemos ver que coincide con el corrimiento de un lugar de $\overline{\beta}$: $(\overline{\beta^3}) = (1, 2, 1)$.

3.1.1 Multiplicación

Ahora veamos como se realiza la multiplicación de γ_1 y $\gamma_2 \in \mathbb{F}_{q^n}$,

$$\gamma_1 \cdot \gamma_2 = (\sum_{i=0}^{n-1} a_i \alpha^{q^i})(\sum_{i=0}^{n-1} b_i \alpha^{q^i}) = \sum_{i=0}^{n-1}\sum_{j=0}^{n-1} a_i b_j \alpha^{q^i} \cdot \alpha^{q^j} \quad \text{donde } a_i, b_j \in \mathbb{F}_q$$

por lo que necesitamos calcular los productos $\alpha^{q^i} \cdot \alpha^{q^j}$. Notemos que si $i \leq j$

$$\begin{aligned} \alpha^{q^i} \cdot \alpha^{q^j} &= \alpha^{q^i + q^j} \\ &= \alpha^{q^i(1+q^{j-i})} \\ &= (\alpha^{1+q^{j-i}})^{q^i} \\ &= (\alpha \cdot \alpha^{q^{j-i}})^{q^i} \end{aligned}$$

entonces sólo es necesario conocer los productos $\alpha \cdot \alpha^{q^k}$, donde $0 \leq k \leq n-1$, en términos de la base BN, ya que los demás productos serán corrimientos de estos.

Definición 3.1.5. *Consideremos los productos* $\alpha \cdot \alpha^{q^i}$

	α	α^q	α^{q^2}	$\cdots$	$\alpha^{q^{n-1}}$
$\alpha \cdot \alpha$	t_{00}	t_{01}	t_{02}	$\cdots$	$t_{0(n-1)}$
$\alpha \cdot \alpha^q$	t_{10}	t_{11}	t_{12}	$\cdots$	$t_{1(n-1)}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\alpha \cdot \alpha^{q^{n-1}}$	$t_{(n-1)0}$	$t_{(n-1)1}$	$t_{(n-1)2}$	$\cdots$	$t_{(n-1)(n-1)}$

donde t_{ij}, es el coeficiente de α^{q^j} en la representación única del producto $\alpha \cdot \alpha^{q^i}$ en términos de la base BN. Sea T_{BN} la matriz formada por los elementos t_{ij}.

Conociendo la matriz T_{BN} se tiene acceso en forma más eficiente a la representación de los productos $\alpha^{q^i}\alpha^{q^j}$. En general como $\alpha^{q^i}\alpha^{q^j} = (\alpha \cdot \alpha^{q^{j-i}})^{q^i}$ la representación de $\alpha^{q^i}\alpha^{q^j}$ será un corrimiento de i lugares a la derecha del renglón $j - i$ de la matriz T_{BN}.

Ejemplo 3.1.6. *Calculemos la matriz T_{BN} para la base BN del ejemplo 3.1.4.*

Obteniendo

T_{BN}	α	α^3	α^{3^2}
$\alpha\alpha$	1	1	2
$\alpha\alpha^3$	0	1	1
$\alpha\alpha^{3^2}$	1	1	0

3.2 Bases Normales Óptimas

En 1988 Mullin y et. al. ([MuOnVa89]) proponen un método para construir bases normales de $\mathbb{F}_{p^n}$ sobre $\mathbb{F}_p$ donde p es un primo, clasificando estas en dos tipos, el *Tipo I* para campos $\mathbb{F}_{p^n}$, y las del *Tipo II* para campos $\mathbb{F}_{2^n}$. Su construcción se realiza a través de teoremas que primero garantizan su existencia bajo ciertas condiciones que debe cumplir n.

3.2.1 Definición de BNO

Consideremos la matriz T_{BN} de una base normal BN del campo $\mathbb{F}_{p^n}$ sobre $\mathbb{F}_p$, dada en la definición 3.1.5. Quisieramos que esta matriz tuviera el mayor número de elementos iguales a cero, ya que esto facilitará la aritmética al

realizar menos operaciones.

Sea C_{BN} el número de elementos distintos de cero de la matriz T_{BN}. Primero veremos que este número es siempre mayor o igual a $2n-1$, donde n es la potencia de p.

Teorema 3.2.1. *Si BN es una base normal de $\mathbb{F}_{p^n}$ sobre $\mathbb{F}_p$ con matriz T_{BN}, entonces $C_{BN} \geq 2n-1$.*

Demostración. Sea $BN = \{\alpha, \alpha^p, \alpha^{p^2}, \ldots, \alpha^{p^{n-1}}\}$ y $\alpha_i = \alpha^{p^i}$ para $1 \leq i < n$. Como BN es una base normal $Tr(\alpha) = \sum_{i=0}^{n-1} \alpha_i = b \neq 0$ donde $b \in \mathbb{F}_p$.

Calculamos $b\alpha_0 = \alpha_0(\sum_{i=0}^{n-1} \alpha_i) = \sum_{i=0}^{n-1} \alpha_0\alpha_i$. Por lo tanto la suma de los renglones de T_{BN} es igual a $b\alpha_0$. Así $b\alpha_0$ se representa por el vector con n entradas

$$(b, 0, \ldots, 0).$$

Entonces hay por lo menos 2 elementos distintos de cero en las columnas $2, \ldots, n$, y por lo menos un componente no nulo en la primera columna, ya que los renglones son linealmente independientes porque BN es una base.

Si sumamos los posibles elementos que son distintos de cero de las columnas obtenemos 1 de la primera más $n-1$ veces 2 de las restantes, dando un total de al menos $1 + 2(n-1) = 1 + 2n - 2 = 2n - 1$ elementos distintos de cero, con lo que

$$C_{BN} \geq 2n - 1.$$

□

Definición 3.2.2. *Una base normal óptima BNO es aquella base normal cuya matriz de productos T_{BNO} tiene exactamente $2n-1$ elementos distintos de cero, es decir, $C_{BN} = 2n-1$.*

Ejemplo 3.2.3. *Sea $\mathbb{F}_{2^3}$ definido como en el ejemplo 1.1.3. La base*

$$BN = \{\beta, \beta^2, \beta^{2^2}\} = \{\alpha+1, \alpha^2+1, \alpha^2+\alpha+1\}$$

es una base normal óptima.

Calculando T_{BN}, obtenemos

T_{BN}	β	β^2	β^{2^2}
$\beta\beta$	0	1	0
$\beta\beta^2$	1	0	1
$\beta\beta^{2^2}$	0	1	1

Como $C_{BN} = 5 = 2(3) - 1 = 2n - 1$, BN es una base normal óptima BNO. La base normal del ejemplo 3.1.6, también fue construida con $n = 3$, pero su matriz T tiene 7 elementos distintos de cero, así que no es una BNO.

Si elegimos apropiadamente a un elemento $\alpha \in \mathbb{F}_{p^n}$, podremos generar una BNO.

3.2.2 Construcción de BNO del *Tipo I*

Primero veremos los siguientes resultados que nos ayudarán a construir una BNO en $\mathbb{F}_{p^n}$.

Teorema 3.2.4. *Supongamos que $\mathbb{F}_{p^n}$ contiene $(n+1)$-ésimas raices de la unidad. Si n raices distintas de 1 son linealmente independientes, entonces $\mathbb{F}_{p^n}$ contiene una BNO sobre $\mathbb{F}_p$.*

Demostración. Sea β una raíz primitiva de la unidad en $\mathbb{F}_{p^n}$, sus conjugados forman la base normal $BN = \{\beta, \beta^p, \ldots, \beta^{p^{n-1}}\}$, ya que son linealmente independientes.

Además notemos que los elementos de la base BN son ceros del polinomio que resulta de la división $p(x) = (x^{n+1} - 1)/(x - 1)$, entonces BN es el conjunto de las raices de la unidad distintas de 1 en $\mathbb{F}_{p^n}$. Sea $\beta_i = \beta^{p^i}$ para, $i = 0, \ldots, n-1$.

Ahora queremos contar el número de elementos distintos de cero de la matriz T_{BN}. Notemos

$$\beta_0\beta_i = \begin{cases} \beta_j \text{ para alguna } j & \text{si } \beta_i \neq \beta_0^{-1} \\ \sum_{k=0}^{n-1} \beta_k & \text{si } \beta_i = \beta_0^{-1}. \end{cases}$$

En el primer caso $n - 1$ coeficientes son distintos de cero en esos renglones, y en el renglón del caso 2 pasa una vez pero contribuye con n coeficientes distintos de cero. Así $C_{BN} = 2n - 1$, y por ende NB es una base normal óptima.

□

Del teorema anterior empezamos a vislumbrar condiciones sobre n, tales que podamos encontrar una BNO contenida en $\mathbb{F}_{p^n}$. Estas condiciones estan dadas por el siguiente teorema, así también la forma de construir una base normal óptima.

Teorema 3.2.5. *Sea el campo $\mathbb{F}_{p^n}$, donde p es primo. $\mathbb{F}_{p^n}$ contiene una BNO formada por las raices $(n+1)$-ésimas de la unidad distintas de 1, si y sólo si, $n+1$ es primo y p es primitivo en $\mathbb{Z}_{n+1}^*$.*

Demostración. Si $n+1$ es primo se cumple que $n+1|p^n-1$ y $\mathbb{F}_{p^n}$ contiene una raíz (n+1)-ésima primitiva de la unidad β. Como p es primitivo en $\mathbb{Z}_{n+1}^*$, el polinomio mínimo de β es $(x^n+x^{n-1}+\cdots+1$ y las raices $(n+1)$-ésimas de la unidad distintas de 1 son linealmente independientes.

Si las raices son linealmente independientes entonces p tiene orden n módulo $n+1$ y así p es primitivo en $\mathbb{Z}_{n+1}^*$ y $n+1$ es primo.

□

Definición 3.2.6 (*BNO* **del** *Tipo I*). *Una base normal óptima BNO de $\mathbb{F}_{p^n}$ sobre $\mathbb{F}_p$ es del Tipo I, si $n+1$ es primo y p es un elemento primitivo en $\mathbb{Z}_{n+1}^*$. La BNO será generada por una raíz β $(n+1)$-ésima primitiva de la unidad en $\mathbb{F}_{p^n}$, siendo los productos*

$$\beta\beta^{p^i} = \begin{cases} \beta^{p^m} \quad \text{para alguna } m & si\ \beta^{p^i} \neq \beta^{-1} \\ 1 = -Tr(\beta) = -\sum_{k=0}^{n-1} \beta^{p^k} & si\ \beta^{p^i} = \beta^{-1} \end{cases}$$

Ejemplo 3.2.7. *Calculemos una base normal óptima para el campo finito $\mathbb{F}_{3^4}$ definido por el polinomio*

$$f(x) = x^4 + x + 3,$$

es decir,

$$\mathbb{F}_{3^4} = \{a_0 + a_1\alpha + a_2\alpha^2 + a_3\alpha^3 : \alpha^4 = 2\alpha + 1, a_i \in \mathbb{F}_3\}.$$

Buscamos una raíz quinta primitiva de la unidad β en $\mathbb{F}_{3^4}$. Se puede ver que $ord(\alpha) = 80$, y que $\beta = \alpha^{16}$ tiene orden 5 por el lema 1.1.6. Entonces β genera una base normal óptima del *Tipo I* ya que $n+1 = 5$ es primo.

Sea

$$B = \{\beta, \beta^3, \beta^{3^2}, \beta^{3^3}\}$$

Donde

$$\begin{aligned} \beta &= \beta_0 = 2\alpha^3 + \alpha + 2 \\ \beta^3 &= \beta_1 = 2\alpha^2 + 2\alpha + 2 \\ \beta^{3^2} &= \beta_2 = 2\alpha^2 + 2 \\ \beta^{3^3} &= \beta_3 = \alpha^3 + 2\alpha^2 + 2 \end{aligned}$$

Calculando los productos $\beta_0\beta_i$ obtenemos:

$$\begin{aligned}\beta_0\beta_0 &= \alpha^3 + 2\alpha^2 + 2 = \beta_3\\ \beta_0\beta_1 &= 2\alpha^2 + 2 = \beta_2\\ \beta_0\beta_2 &= 1 = -(\beta_0 + \beta_1 + \beta_2 + \beta_3)\\ \beta_0\beta_3 &= 2\alpha^2 + 2\alpha + 2 = \beta_1\end{aligned}$$

y

T_B	β_0	β_1	β_2	β_3
$\beta_0\beta_0$	0	0	0	1
$\beta_0\beta_1$	0	0	1	0
$\beta_0\beta_2$	-1	-1	-1	-1
$\beta_0\beta_3$	1	0	0	0

$C_B = 7$, entonces claramente B es una base normal óptima del *Tipo I*

La construcción en el teorema 3.2.5 nos sirve para encontrar una *BNO* de $\mathbb{F}_{p^n}$ excepto cuando n es primo ($n \neq 2$) puesto que $n+1$ no es primo. Esta construcción no es única, en el caso cuando $p = 2$ se pueden encontrar *BNO* de $\mathbb{F}_{2^n}$ sobre $\mathbb{F}_2$ bajo otras condiciones. Veremos este tipo llamado *Tipo II* en la siguiente sección.

3.2.3 Construcción de *BNO* del *Tipo II*

Al igual que con las bases normales óptimas del *Tipo I*, empezaremos con un resultado que nos dará una forma para construir *BNO* del *Tipo II*.

Teorema 3.2.8. *Sea n un entero positivo. Entonces el campo finito $\mathbb{F}_{2^n}$ contiene una BNO si*

(a) $< 2 >= \{2^i : 0 \leq i \leq 2n-1\} = \mathbb{Z}^*_{2n+1}$,

ó

(b) $2n + 1 \equiv 3 \bmod 4$ *y* 2 *genera los residuos cuadráticos de* $\mathbb{Z}^*_{2n+1}$.

Demostración. (a) Como 2 genera a $\mathbb{Z}^*_{2n+1}$, tiene orden $2n$, y así $2^{2n} \equiv 1 \bmod (2n+1)$. Por lo tanto existe una raíz $(2n+1)$-ésima de la unidad $\beta \in \mathbb{F}_{2^{2n}}$.

Si $2^n \equiv -1 \bmod (2n+1)$ se cumple que $\beta^{2^n} = \beta^{-1}$.

Sea

$$\gamma = \beta + \beta^{-1}, \tag{3.1}$$

puesto que

$$\gamma^{2^n} = (\beta + \beta^{-1})^{2^n} = \beta^{2^n} + (\beta^{-1})^{2^n} = \beta^{-1} + \beta = \gamma$$

por el lema 1.2.4 se tiene que $\gamma \in \mathbb{F}_{2^n}$. Ahora se afirma que

$$BN = \{\gamma, \gamma^2, \ldots, \gamma^{2^{n-1}}\},$$

es una base normal óptima. En efecto, sus elementos son linealmente independientes ya que

$$\begin{aligned} 0 = \sum_{i=0}^{n-1} \lambda_i \gamma^{2^i} &= \sum_{i=0}^{n-1} \lambda_i (\beta^{2^i} + \beta^{-2^i}) \qquad \text{donde } \lambda_i \in \mathbb{F}_2 \\ &= \sum_{i=0}^{n-1} \lambda_i \beta^{2^i} + \sum_{i=0}^{n-1} \lambda_i \beta^{-2^i} \\ &= \sum_{i=0}^{n-1} \lambda_i \beta^{2^i} + \sum_{i=n}^{2n-1} \lambda_{i-n} \beta^{2^i} \\ &\quad \text{ya que } -2^{i-n} = -2^i/2^n = -2^i/(-1) \equiv 2^i \bmod 2n+1 \\ &= \sum_{j=0}^{2n-1} c_j \beta^{2^j}, \quad \text{donde } c_j = \lambda_i \text{ para alguna } i. \end{aligned}$$

Siendo β un cero del polinomio

$$f(x) = \sum_{j=0}^{2n-1} c_j x^j, \tag{3.2}$$

este es dividido por el polinomio mínimo de β

$$m_\beta(x) = 1 + x + x^2 + \cdots + x^{2n}, \quad \text{ya que } < 2 >= \mathbb{Z}^*_{2n+1},$$

como $m_\beta(x)$ tiene grado $2n$ y divide a $f(x)$ que tiene grado $2n-1$, entonces $f(x) \equiv 0$, así $\lambda_i = 0$ para $0 \leq i \leq$.

Ahora contemos los elementos distintos de cero de T_{BN},

$$\begin{aligned}\gamma^{2^i}\gamma^{2^j} &= (\beta^{2^i+\beta^{-2^i}})(\beta^{2^j+\beta^{-2^j}}) \\ &= (\beta^{(2^i+2^j)} + \beta^{-(2^i+2^j)}) + (\beta^{(2^i-2^j)} + \beta^{-(2^i-2^j)})\end{aligned}$$

Como $\mathbb{Z}^*_{2n+1} = \{2^i : 0 \leq i \leq 2n-1\}$, existen $0 \leq k, k' \leq 2n-1$ tales que

$$\gamma^{2^i}\gamma^{2^j} = \begin{cases} \gamma^{2^k} + \gamma^{2^{k'}} & \text{si } 2^i \neq 2^j \bmod 2n+1 \\ \gamma^{2^k} & \text{si } 2^i \equiv \pm 2^j \bmod 2n+1. \end{cases}$$

Por lo tanto son $2n-1$ los elementos distintos de cero en la matriz de productos T_{BN}.

(b) Como $2n+1$ es primo $2^{2n} \equiv 1 \bmod 2n+1$. Entonces existe una raíz $2n+1$-ésima primitiva de la unidad β en $\mathbb{F}_{2^{2n}}$. Sea $\gamma = \beta + \beta^{-1}$. En este caso como 2 genera a los residuos cuadráticos de $\mathbb{Z}^*_{2n+1}$, entonces, $2^n \equiv 1 \bmod 2n+1$ y $\gamma^{2^n} = \beta^{2^n} + (\beta^{-1})^{2^n} = \beta + \beta^{-1}$ por lo tanto $\gamma \in \mathbb{F}_2^n$.

En este caso afirmamos también que los conjugados de γ generan una base normal de $\mathbb{F}_{2^n}$ sobre $\mathbb{F}_2$. Usamos la combinación lineal definida en el caso anterior y el polinomio (3.2) (puesto que tiene a β como raíz). El polinomio mínimo de β $m_\beta(x)$ tiene grado n, también el polinomio mínimo de β^{-1}, $m_{\beta^{-1}}(x)$ tiene grado n. Como polinomios mínimos ambos dividen a $f(x)$ ya que $f(\beta) = f(\beta^{-1}) = 0$ obteniendo que $1 + x + x^2 + \cdots + x^{2n} | f(x)$ con lo que $\lambda_i = 0$ para $0 \leq i \leq n-1$.

Finalmente como $2n+1 \equiv 3 \bmod 4$, se cumple

$$\mathbb{Z}^*_{2n+1} = \{2^i : 0 \leq i \leq n-1\} \cup \{-2^i : 0 \leq i \leq n-1\}$$

Entonces existen enteros $0 \leq k, k' \leq n-1$ tales que

$$2^i + 2^j = \pm 2^k \qquad \text{y} \qquad 2^i - 2^j = \pm 2^{k'}.$$

Por lo que se cumple alguno de los casos:

$$\gamma^{2^i}\gamma^{2^j} = \begin{cases} \gamma^{2^k} + \gamma^{2^{k'}} & \text{si } 2^i \neq 2^j \bmod 2n+1 \\ \gamma^{2^k} & \text{si } 2^i \equiv \pm 2^j \bmod 2n+1. \end{cases}$$

y así también en este caso se genera una base normal óptima.

□

Definición 3.2.9 (*BNO* **del** *Tipo II*). *Una base normal óptima es de Tipo II si satisface las condiciones del teorema anterior.*

Ejemplo 3.2.10. *Sea* $n = 3$, $2n+1 = 7 \equiv 3 \, mod 4$ *y* 2 *genera a los residuos cuadráticos de* $\mathbb{Z}_7^*$. *Así podemos encontrar una raíz séptima primitiva de la unidad en* $\mathbb{F}_{2(2)(3)} = \mathbb{F}_{2^6}$. *Definamos las operaciones sobre este campo utilizando el polinomio irreducible* $f(x) = x^6 + x + 1$, *es decir,*

$$\mathbb{F}_{2^6} = \{\sum_{i=0}^{5} a_i \alpha^i : \alpha^6 = \alpha + 1 \; a_i \in \mathbb{F}_2\}$$

El elemento $\beta = \alpha^9 = \alpha^4 + \alpha^3$, es raíz séptima de la unidad de $\mathbb{F}_{2^6}$. Se puede ver facilmente que $\beta^{-1} = \alpha^4 + \alpha^2 + \alpha + 1$ y así

$$\gamma = \beta + \beta^{-1} = \alpha^3 + \alpha^2 + \alpha + 1$$

Considerando $\gamma_i = \gamma^{2^i}$, como 2 genera a los residuos cuadráticos de $\mathbb{Z}_7^*$ y $7 \equiv 3 \bmod 4$, se tiene que $B = \{\gamma_0, \gamma_1, \gamma_2\}$ es una base normal del *Tipo II* caso (b), donde

$$\begin{aligned} \gamma_0 &= \alpha^3 + \alpha^2 + \alpha + 1 \\ \gamma_1 &= \alpha^4 + \alpha^2 + \alpha \\ \gamma_2 &= \alpha^4 + \alpha^3 \end{aligned}$$

con la siguiente tabla de productos

T_B	γ_0	γ_1	γ_2
$\gamma_0\gamma_0$	0	1	0
$\gamma_0\gamma_1$	1	0	1
$\gamma_0\gamma_2$	0	1	1

Contando las entradas de la matriz distintas de cero, tenemos que es 5. Por lo tanto B es una base normal óptima.

Ejemplo 3.2.11. *Ahora consideremos* $n = 5$. *Se tiene que* $< 2 >= \mathbb{Z}_{11}^*$ *y usando el polinomio* $f(x) = x^{10} + x^3 + 1$ *irreducible sobre* $\mathbb{F}_2[x]$, *tenemos*

$$\mathbb{F}_{2^{10}} = \{\sum_{i=0}^{9} a_i \alpha^i : \alpha^{10} = \alpha^3 + 1 \; a_i \in \mathbb{F}_2\}.$$

El elemento $\beta = \alpha^9 + \alpha^7 + \alpha^6 + \alpha^2 + 1$ es una raíz 11-ésima de la unidad. Calculando su inversa obtenemos $\beta^{-1} = \alpha^8 + \alpha^7 + \alpha^6 + \alpha^3 + \alpha + 1$. Con β y β^{-1} sea

$$\gamma = \beta + \beta^{-1} = \alpha^9 + \alpha^8 + \alpha,$$

$\gamma_i = \gamma^{2^i}$ y

$$B = \{\gamma_0, \gamma_1, \gamma_2, \gamma_3, \gamma_4\}$$

es una *BNO* del *Tipo II* caso (b), donde

$$\begin{aligned}
\gamma_0 &= \alpha^9 + \alpha^8 + \alpha \\
\gamma_1 &= \alpha^9 + \alpha^8 + \alpha^6 + \alpha^4 + \alpha^2 + \alpha \\
\gamma_2 &= \alpha^9 + \alpha^6 + \alpha^5 + \alpha \\
\gamma_3 &= \alpha^8 + \alpha^5 + \alpha^4 + \alpha^3 + \alpha + 1 \\
\gamma_4 &= \alpha^9 + \alpha^8 + \alpha^3 + \alpha^2
\end{aligned}$$

con la siguiente tabla de productos

T_B	γ_0	γ_1	γ_2	γ_3	γ_4
$\gamma_0\gamma_0$	0	1	0	0	0
$\gamma_0\gamma_1$	1	0	0	1	0
$\gamma_0\gamma_2$	0	0	0	1	1
$\gamma_0\gamma_3$	0	1	1	0	0
$\gamma_0\gamma_4$	0	0	1	0	1

Se puede observar que el número de elementos distintos de cero es 9.

El siguiente lema nos ayudará a construir este tipo de bases.

Lema 3.2.12. *Sea p y q primos distintos entonces:*

(a) 2 es primitivo en $\mathbb{Z}_p^*$, si $p = 4q + 1$ y q es impar,

(b) 2 es primitivo en $\mathbb{Z}_p^*$, si $p = 2q + 1$ y $q \equiv 1 \bmod 4$, y

(c) -2 es primitivo en $\mathbb{Z}_p^*$, si $p = 2q + 1$ donde $q \equiv 3 \bmod 4$.

Demostración. Usando el teorema 3.3 de [NiZu85] 2 es residuo cuadrático si $(-1)^{\frac{p^2-1}{8}} = 1$. Entonces los casos se siguen aplicando esta fórmula directamente.

□

2	3	4	5	6	9	10	11	12	14	18	23
26	28	29	30	33	35	36	39	41	50	51	52
53	58	60	65	66	69	74	81	82	83	86	89
90	95	98	99	100	105	106	113	119	130	131	134
135	138	146	148	155	158	162	172	173	174	178	179
180	183	186	189	191	194	196	209	210	221	226	230
231	233	239	243	245	251	254	261	268	270	273	278
281	292	293	299	303	306	309	316	323	326	329	330
338	346	348	350	354	359	371	372	375	378	386	388
393	398	410	411	413	414	418	419	420	426	429	431
438	441	442	443	453	460	466	470	473	483	490	491
495	508	509	515	519	522	530	531	540	543	545	546
554	556	558	561	562	575	585	586	593	606	611	612
614	615	618	629	638	639	641	645	650	651	652	653
658	659	660	676	683	686	690	700	708	713	719	723
725	726	741	743	746	749	755	756	761	765	771	772
774	779	783	785	786	791	796	803	809	810	818	820
826	828	831	833	834	846	852	858	866	870	873	876
879	882	891	893	906	911	923	930	933	935	938	989
940	946	950	953	965	974	975	986	989	993	998	1013
1014	1018	1019	1026	1031	1034	1041	1043	1049	1055	1060	1065
1070	1090	1103	1106	1108	1110	1116	1118	1119	1121	1122	1133
1134	1146	1154	1155	1166	1169	1170	1178	1185	1186	1194	1199

Tabla 3.1: Valores $n < 1200$ para los que se puede construir una *BNO*

Con el lema anterior podemos saber fácilmente si 2 es primitivo módulo $2n+1$, cuando n cumple alguna de las condiciones mencionadas.

En la tabla 3.1 [MuOnVa89] estan valores n tales que $n+1$ es primo o $2n+1$ es primo, para construir bases normales óptimas de *Tipo I* o de *Tipo II* respectivamente.

3.3 Períodos de Gauss

Carl Friedrich Gauss (1777-1855), en su libro *Disquisitiones arithmeticae* publicado en 1801, en los artículos 343-366, introduce un método que utilizaremos para construir bases normales. Gauss desarrolló esta idea para investigar cuando un polígono regular puede ser construido con regla y compás [Ga86]. Los períodos de Gauss fueron originalmente definidos para campos numéricos, lo que veremos es una adaptación para campos finitos. Los

períodos de Gauss, en ciertos casos generan una base normal del campo finito $\mathbb{F}_{q^n}$ sobre $\mathbb{F}_q$, donde $q = p^m$ con p primo. El uso de estas bases para realizar aritmética fue propuesto en [GaGaPa95, GaGaPa98, GaGaPaSh].

3.3.1 Definición

Definición 3.3.1 (Par de **Gauss**). *Sea $\mathbb{F}_{q^n}$ un campo finito. Se dice que (n,k) es un par de Gauss, si k es un entero tal que $nk+1$ es primo y $(q, nk+1) = 1$.*

Ejemplo 3.3.2. *Sobre el campo finito $\mathbb{F}_{3^2}$, $(2,5)$ es un par de Gauss ya que $(3,11) = 1$ y $2(5)+1 = 11$ es primo.*

Si $nk+1$ es primo y $(nk+1, q) = 1$ entonces se cumple que $q^{nk} \equiv$ mod $nk+1$, por lo que $nk+1 | q^{nk} - 1$. Por tal razón existe una raíz $(nk+1)$-ésima de la unidad β en $\mathbb{F}_{q^{nk}}$.

Lema 3.3.3. *Si $nk+1$ es primo entonces $\mathbb{Z}^*_{nk+1}$ tiene un único subgrupo de orden k.*

Demostración. Sabemos que $\mathbb{Z}^*_{nk+1}$ es un grupo cíclico de orden nk. Como $k | nk$, existe un subgrupo de $\mathbb{Z}^*_{nk+1}$ de orden k, digamos $\mathcal{K}$. Siendo $\mathcal{K}$ un subgrupo de un grupo cíclico, es cíclico a su vez. Sea g el generador de $\mathcal{K}$, por lo que $\text{ord}(g) = k$. Por otro lado, $\mathbb{Z}^*_{nk+1}$ contiene $\phi(k)$ elementos de orden k, donde $\phi(k)$ es la función ϕ de Euler, los cuales son precisamente las potencias de g que son primos relativos a k, las cuales estan contenidas en $\mathcal{K}$. Por lo tanto $\mathcal{K}$ es el único subgrupo de orden k.

□

Definimos las clases laterales del subgrupo $\mathcal{K}$ como $\mathcal{K}_i$ para $0 \leq i \leq n-1$, de la siguiente forma

$$\mathcal{K}_i = \{aq^i : a \in \mathcal{K}\}.$$

Definición 3.3.4 (Períodos de Gauss del tipo (n,k)). *Sea n, q, k enteros tales que $(nk+1, q) = 1$, $nk+1$ es primo, β es una raíz $(nk+1)$-ésima de la unidad en $\mathbb{F}_{q^{nk}}$, $\mathcal{K} \subset \mathbb{Z}^*_{nk+1}$ el único subgrupo de orden k y las clases laterales $\mathcal{K}_i$ descritas antes. Entonces los períodos de Gauss para $0 \leq i \leq n-1$ son:*

$$\alpha_i = \sum_{a \in \mathcal{K}_i} \beta^a.$$

Lema 3.3.5. *Sean los períodos de Gauss* $\{\alpha_0, \alpha_1, \ldots, \alpha_{n-1}\}$ *como en la definición 3.3.4. Entonces* $\alpha_i \in \mathbb{F}_{q^n}$ *para* $0 \leq i \leq n-1$.

Demostración. Por el lema 1.2.4, el período de Gauss α_0 pertenece a $\mathbb{F}_{q^n}$ si $(\alpha_0)^{q^n} = \alpha_0$.

Calculando

$$\begin{aligned}(\alpha_0)^{q^n} &= (\sum_{a\in\mathcal{K}} \beta^a)^{q^n} = \sum_{a\in\mathcal{K}} (\beta^a)^{q^n} \\ &= \sum_{a\in\mathcal{K}} \beta^{aq^n} = \sum_{b\in q^n\mathcal{K}} \beta^b\end{aligned}$$

Nos resta demostrar que las clases $\mathcal{K}$, $q^n\mathcal{K}$ son iguales. En efecto, $(q^n)^k = 1$ por lo tanto $\mathrm{ord}(q^n)|k$ con lo que $q^n \in \mathcal{K}$. Entonces $q^n\mathcal{K} = \mathcal{K}$ y

$$(\alpha_0)^{q^n} = \sum_{b\in q^n\mathcal{K}} \beta^b = \sum_{a\in\mathcal{K}} \beta^a = \alpha_0$$

De la misma forma $(\alpha_i)^{q^n} = \alpha_i$ ya que $q^n\mathcal{K}_i = \mathcal{K}_i$ para $1 \leq i \leq n-1$.

□

Teorema 3.3.6. *Sea* $\mathbb{F}_{q^n}$ *un campo finito,* (n,k) *un par de Gauss, e el orden de* q *módulo* $nk+1$, $\mathcal{K}$ *el único subgrupo de orden* k *del grupo* $\mathbb{Z}^*_{nk+1}$ *y* β *una raíz* $(nk+1)$*-ésima primitiva de la unidad en* $\mathbb{F}_{q^{nk}}$. *Entonces el período de Gauss*

$$\alpha = \sum_{a\in\mathcal{K}} \beta^a,$$

genera una una base normal de $\mathbb{F}_{q^n}$ *sobre* $\mathbb{F}_q$ *si y sólo si* $(nk/e, n) = 1$.

Demostración. Definamos para $0 \leq i \leq e-1$,

$$\mathcal{K}_i = \{aq^i : a \in \mathcal{K}\} \subseteq \mathbb{Z}^*_{nk+1}, \quad \alpha_i = \sum_{a\in\mathcal{K}_i} \beta^a. \tag{3.3}$$

Con $\mathcal{K}_0 = \mathcal{K}$ y el período de Gauss $\alpha_0 = \alpha$. Tenemos además que

$$\mathcal{K}_i = \{aq^i : a \in \mathcal{K}\} = \{aq^{i-1} : a \in \mathcal{K}\}q = \mathcal{K}_{i-1}q = \mathcal{K}_{i-2}q^2 = \mathcal{K}q^i,$$

y

$$\alpha^{q^i} = (\sum_{a\in\mathcal{K}} \beta^a)^{q^i} = \sum_{a\in\mathcal{K}} (\beta^a)^{q^i} = \sum_{a\in\mathcal{K}} \beta^{aq^i} = \sum_{b\in\mathcal{K}_i} \beta^b = \alpha_i.$$

Sea $i_{\mathcal{K}} = n$ el índice del subgrupo cíclico $\mathcal{K}$, e $i_q = nk/e$ el índice del subgrupo generado por q módulo $nk+1$, donde e es el orden de q, ambos índices con respecto al grupo $\mathbb{Z}^*_{nk+1}$. El índice del grupo formado por

$$\mathcal{K} < q >= \{aq^i : a \in \mathcal{K} \text{ y } 0 \leq i < e\},$$

es $(i_{\mathcal{K}}, i_q)$, entonces

$$(nk/e, n) = 1 \iff \mathbb{Z}^*_{nk+1} = \mathcal{K} < q > . \tag{3.4}$$

Esta condición es equivalente a que el período de Gauss α genera una base normal de $\mathbb{F}_{q^n}$ sobre $\mathbb{F}_q$ puesto que si consideramos una combinación lineal

$$\sum_{i=0}^{n-1} \lambda \alpha^{q^i} = 0.$$

La condición 3.4 asegura que todas las clases $\mathcal{K}_i$ son ajenas y distintas. Entonces:

$$\sum_{i=0}^{n-1} \lambda \alpha^{q^i} = \sum_{a \in \mathbb{Z}} c_a \beta^a = 0,$$

donde $c_a = \lambda_i$ para alguna $0 \leq i \leq n-1$. Como son todas la raices primitivas β y estas son linealmente independientes, $c_a = 0$ por lo tanto las $\lambda_i = 0$.

Entonces $BNPG = \{\alpha_0, \alpha_1, \ldots, \alpha_{n-1}\}$ es una base normal de $\mathbb{F}_{q^n}$ sobre $\mathbb{F}_q$.

□

De aquí en adelante supondremos que $nk+1$ es primo y $(nk/e, n) = 1$, donde e es el orden de q módulo $nk+1$.

Ejemplo 3.3.7. *Consideremos el campo finito $\mathbb{F}_{3^5}$, es decir, $n = 5$, $q = 3$, para $k = 2$, $nk+1 = 11$ es primo, $e = ord(q) = 5$. Como se cumple que $(10/5, 5) = (2, 5) = 1$, sea β una raíz 11-ésima primitiva de la unidad en $\mathbb{F}_{3^{10}}$. Tenemos*

$$\mathbb{Z}^*_{11} = \{1,2,3,4,5,6,7,8,9,10\}, \quad < 3 >= \{3,9,5,4,1\} \; y \quad \mathcal{K} = \{1,10\}$$

$$\mathcal{K}_1 = \{3,8\} \quad \mathcal{K}_2 = \{9,2\} \quad \mathcal{K}_3 = \{5,6\} \quad \mathcal{K}_4 = \{4,7\}$$

Entonces la base normal es

$$\{\alpha_0, \alpha_1, \alpha_2, \alpha_3, \alpha_4\} = \{\beta + \beta^{10}, \beta^3 + \beta^8, \beta^9 + \beta^2, \beta^5 + \beta^6, \beta^4 + \beta^5\} \quad ,$$

con $\beta^{11} = 1$.

La suma de elementos representados en términos de una base normal generada por un período de Gauss, es coeficiente a coeficiente como ya hemos visto.

3.3.2 Multiplicación

Para multiplicar elementos representados en términos de la base normal generada por los períodos de Gauss $\alpha, \alpha_1, \ldots, \alpha_{n-1}$, recordamos de la sección 3.1.1 que tenemos que conocer los productos $\alpha\alpha_i$ para $0 \leq i \leq n-1$. Las herramientas que utilizaremos son las siguientes.

Sea t_{ij} (en la teoría de campos ciclotómicos estos números son llamados números ciclotómicos [Wa82]) el número de elementos $a \in \mathcal{K}_j$ tales que, $1 + a \in \mathcal{K}_i$, es decir,

$$t_{ij} = |(1 + \mathcal{K}_i) \cap \mathcal{K}_j| \quad \text{para } 0 \leq i, j \leq n-1.$$

Sea $i_0 < n$ el único subíndice tal que $nk \in \mathcal{K}_{i_0}$ (es único puesto que las clases laterales $\mathcal{K}_i$ son ajenas y su unión es todo $\mathbb{Z}^*_{nk+1}$).

Lema 3.3.8.

$$\sum_{j=0}^{n-1} t_{ij} = \begin{cases} k, & si\ i \neq i_0 \\ k-1, & si\ i = i_0 \end{cases}$$

Demostración. Como las clases $\mathcal{K}_j$ son ajenas, calculamos

$$\begin{aligned} \sum_{j=0}^{n-1} t_{ij} &= \sum_{j=0}^{n-1} |(1 + \mathcal{K}_i) \cap \mathcal{K}_j| \\ &= |\cup_{j=0}^{n-1} ((1 + \mathcal{K}_i) \cap \mathcal{K}_j)| \\ &= |(1 + \mathcal{K}_i) \cap (\cup_{j=0}^{n-1} \mathcal{K}_j)| \\ &= |(1 + \mathcal{K}_i) \cap \mathbb{Z}^*_{nk+1}| \end{aligned}$$

Si $i \neq i_0$, entonces $0 \notin 1 + \mathcal{K}_i$ y $(1 + \mathcal{K}_i) \subset \mathbb{Z}^*_{nk+1}$, por lo que la cardinalidad de la intersección es la del conjunto $1 + \mathcal{K}_i$. Si $i = i_0$, entonces $0 \in 1 + \mathcal{K}_i$ y $(1 + \mathcal{K}_i - \{0\}) \subset \mathbb{Z}^*_{nk+1}$, de esta forma la intersección tiene cardinalidad $k-1$.

□

Teorema 3.3.9. *Para* $0 \leq i \leq n-1$,

$$\alpha\alpha_i = \begin{cases} \sum_{0\leq j\leq n-1} t_{ij}\alpha_j, & si\ i \neq i_0 \\ \sum_{0\leq j\leq n-1} t_{ij}\alpha_j + k, & si\ i = i_0. \end{cases} \tag{3.5}$$

Demostración. Calculamos $\alpha\alpha_i$ directamente:

$$\begin{aligned} \alpha\alpha_i &= \left(\sum_{a\in\mathcal{K}} \beta^a\right)\left(\sum_{b\in\mathcal{K}} \beta^{bq^i}\right) = \sum_{a,b\in\mathcal{K}} \beta^{a+bq^i} \\ &= \sum_{a,b\in\mathcal{K}} \beta^{a(1+bq^i)} = \sum_{b\in\mathcal{K}}\sum_{a\in\mathcal{K}} \beta^{a(1+bq^i)}. \end{aligned}$$

Para cada $b \in \mathcal{K}$ tenemos que $1 + bq^i \equiv 0 \bmod nk+1$, o $1 + bq^i \in \mathcal{K}_j$ para una única j con $0 \leq j \leq n-1$.

Si $1 + bq^i \equiv 0 \bmod nk+1$ entonces:

$$\sum_{a\in\mathcal{K}} \beta^{a(1+bq^i)} = k,$$

donde k es el orden de $\mathcal{K}$. Este caso pasa cuando $nk \in \mathcal{K}_i$, es decir, el subíndice especial i_0.

Si $1 + bq^i \in \mathcal{K}_j$, entonces:

$$\sum_{a\in\mathcal{K}} \beta^{a(1+bq^i)} = \sum_{a\in\mathcal{K}_j} \beta^a = \alpha_j.$$

Esto sucede cuando $b \in \mathcal{K}_j$, por lo que precisamente obtendremos α_j tantas veces como $b \in 1 + \mathcal{K}_j$ que es precisamente t_{ij}.

□

Los t_{ij} para $i \neq i_0$, son las entradas de la matriz T_{BNPG} (definición 3.1.5) formada por los productos $\alpha\alpha_i$. Como el conjunto $1 + \mathcal{K}_i$ tiene k elementos distintos de cero si $i \neq i_0$ y tiene $k-1$ si $i = i_0$ entonces hay a lo más k entradas t_{ij} distintas de cero para cualquier subíndice i fijo. Así la matriz T_{BNPG} tiene a lo más nk elementos distintos de cero, incluyendo los k elementos en $\alpha\alpha_{i_0}$. Por esta razón queremos encontrar una base normal para k pequeño.

Ejemplo 3.3.10. *Consideremos el campo* $\mathbb{F}_{3^5}$ *y la base normal generada como en el ejemplo 3.3.7. Como*

$$1+\mathcal{K}_0=\{2,0\} \quad 1+\mathcal{K}_1=\{4,9\} \quad 1+\mathcal{K}_2=\{3,10\}$$

$$1+\mathcal{K}_3=\{6,7\} \quad 1+\mathcal{K}_4=\{5,8\}$$

Se tiene que:

$$\begin{array}{lllll} t_{00}=0 & t_{01}=0 & t_{02}=1 & t_{03}=0 & t_{04}=0 \\ t_{10}=0 & t_{11}=0 & t_{12}=1 & t_{13}=0 & t_{14}=1 \\ t_{20}=1 & t_{21}=1 & t_{22}=0 & t_{23}=0 & t_{24}=0 \\ t_{30}=0 & t_{31}=0 & t_{32}=0 & t_{33}=1 & t_{34}=1 \\ t_{40}=0 & t_{41}=1 & t_{42}=0 & t_{43}=1 & t_{44}=0 \end{array}$$

Obteniendo

$$\begin{aligned} \alpha\alpha &= \alpha_2+2 \\ \alpha\alpha_1 &= \alpha_2+\alpha_4 \\ \alpha\alpha_2 &= \alpha+\alpha_1 \\ \alpha\alpha_3 &= \alpha_3+\alpha_4 \\ \alpha\alpha_4 &= \alpha_1+\alpha_3 \end{aligned}$$

Puesto que las bases autoduales son útiles en la implementación [Wa89], veremos cuando las bases normales generadas por los períodos de Gauss son autoduales.

Teorema 3.3.11. *Sea* α_i, i_0 *como en el Teorema 3.3.9. Definamos*

$$\gamma=\frac{\alpha_{-i_0}-k}{nk+1}.$$

Entonces $\{\gamma,\gamma^q,\ldots,\gamma^{q^{n-1}}\}$ *es la base dual de* $\{\alpha_0,\alpha_1,\ldots,\alpha_{n-1}\}$.

Demostración. Sabemos que $Tr(\alpha_i)=-1$ para $0\leq i\leq n-1$.

$$\begin{aligned} &Tr(\gamma^{q^i}\alpha_j)=Tr((\frac{\alpha_{-i_0}-k}{nk+1})^{q^i}\alpha_j) \\ &=Tr(\frac{(\alpha_{-i_0})^{q^i}-k}{nk+1}\alpha_j) \\ &=Tr(\frac{\alpha_{-i_0+i}\alpha_j-k\alpha_j}{nk+1}) \end{aligned}$$

$$= \frac{Tr((\alpha\alpha_{j-i+i_0})^{q^{-i_0+i}} - k\alpha_j)}{nk+1}$$
$$= \frac{Tr((\alpha\alpha_{j-i+i_0})^{q^{-i_0+i}}) - kTr(\alpha_j)}{nk+1}$$
$$= \frac{Tr(\alpha\alpha_{j-i+i_0}) - k(-1))}{nk+1} \quad \text{por el teorema 3.3.9}$$
$$= \begin{cases} \dfrac{Tr(\sum_{l=0}^{n-1} t_{il}\alpha_l) + k}{nk+1}, & \text{si } l = j-i+i_0 \neq i_0 \\ \dfrac{Tr(\sum_{l=0}^{n-1}(t_{il}\alpha_l) + k) + k)}{nk+1}, & \text{si } l = j-i+i_0 = i_0 \end{cases}$$

Por propiedades de la traza

$$= \begin{cases} \dfrac{-(\sum_{l=0}^{n-1} t_{il}) + k}{nk+1}, & \text{si } j \neq i \\ \dfrac{-(\sum_{l=0}^{n-1}(t_{il}) + nk + k)}{nk+1}, & \text{si } j = i \end{cases}$$

Por el lema 3.3.8

$$= \begin{cases} \dfrac{-(k) + k}{nk+1}, & \text{si } j \neq i \\ \dfrac{-(k-1) + nk + k)}{nk+1}, & \text{si } j = i \end{cases}$$
$$= \begin{cases} 0, & \text{si } j \neq i \\ 1, & \text{si } j = i \end{cases}$$

Por lo tanto la base $\{\gamma, \gamma^q, \ldots, \gamma^{q^{n-1}}\}$ es dual a $\{\alpha_0, \alpha_1, \ldots, \alpha_{n-1}\}$.

□

Corolario 3.3.12. *Para $n > 2$, la base normal BNGP generada por el período de Gauss, construido por el par de Gauss (n,k), es autodual si y sólo si k es par y es divisible por la característica de $\mathbb{F}_q$.*

Demostración. Sabemos que

$$\gamma = \frac{\alpha_{-i_0} - k}{nk+1} = \frac{1}{nk+1}\alpha_{-i_0} + \sum_{l=0}^{n-1} \frac{k}{nk+1}\alpha_l.$$

Entonces $\gamma = \alpha$ si y sólo si $i_0 = 0$ y k es divisible por la característica de $\mathbb{F}_q$. Pero $i_0 = 0$ si y sólo si $nk = -1 \in \mathcal{K}$, esto es, si y sólo si el orden de $\mathcal{K}$ es par, es decir, k es par.

□

Ejemplo 3.3.13. *Nuevamente consideremos la base normal BNPG del ejemplo 3.3.7. Vemos que su base dual de acuerdo con el teorema 3.3.11, tiene como generador a:*

$$\gamma = \frac{\alpha_{-0} - 2}{11} = 2\alpha + 2,$$

225993

entonces BNPG no es autodual. Calculamos las trazas

$$\begin{aligned}
Tr(\alpha\gamma) &= Tr(2\alpha^2 + 2\alpha) = 2Tr(\alpha^2) + Tr(2) = 2(0) + 1 = 1 \\
Tr(\alpha\gamma^3) &= Tr(2\alpha\alpha_1) + Tr(2\alpha) = 2Tr(\alpha_2 + \alpha_4) + 2(-1) = 0 \\
Tr(\alpha\gamma^{3^2}) &= Tr(2\alpha\alpha_2) + Tr(2\alpha) = 2Tr(\alpha + \alpha_1) + 2(-1) = 0 \\
Tr(\alpha\gamma^{3^3}) &= Tr(2\alpha\alpha_3) + Tr(2\alpha) = 2Tr(\alpha_3 + \alpha_4) + 2(-1) = 0 \\
Tr(\alpha\gamma^{3^4}) &= Tr(2\alpha\alpha_4) + Tr(2\alpha) = 2Tr(\alpha_1 + \alpha_3) + 2(-1) = 0
\end{aligned}$$

No necesitamos calcular la traza de todos los productos, puesto que

$$Tr((\alpha\gamma^{3^i})^{q^j}) = Tr(\alpha\gamma^{3^i})$$

para $0 \leq i \leq 4$ *y cualquier* j.

Proposición 3.3.14. *Las condiciones para construir bases normales generadas por períodos de Gauss sobre* $\mathbb{F}_{2^n}$ *son equivalentes a las condiciones para construir bases normales óptimas de (Tipo II) (ver sección 3.2).*

Demostración. $\Longrightarrow$) $2n+1$ es primo y $(\frac{2n}{\text{ord}(2)}, n) = 1$ como $\text{ord}(2)|2n$, los posibles divisores $\{id : i = 1, 2, d|n\}$.

- Si consideramos $\text{ord}(2) = 2$ entonces $(\frac{2n}{2}, n) = n$, lo cual no puede ser ya que $n \neq 1$.
- Si $\text{ord}(2) = d$ es un divisor de n tal que $d \neq n$ tenemos que

  $$n = r_1 d \text{ para } r_1 \in \mathbb{Z} \implies (\frac{2r_1 d}{d}, r_1 d) = (2r_1, r_1 d) \leq r_1 \neq 1$$

  por lo que tampoco pasa,
- y los divisores de la forma $2d|2n$ son equivalentes a los $d|n$.

Entonces los únicos ordenes posibles son n y $2n$.

(a) Si el orden de 2 es $2n$, entonces $< 2 >= \mathbb{Z}^*_{2n+1}$

(b) Si el orden de 2 es n, entonces $(\frac{2n}{n}, n) = (2, n) = 1 \Longrightarrow n$ es impar, es decir, es de la forma $2s+1$, por lo que $2(2s+1)+1 = 4s+3$ y 2 genera los residuos cuadráticos de $\mathbb{Z}^*_p$).

$\Longleftarrow$)

(a) Si $< 2 >= \mathbb{Z}^*_{2n+1}$, entonces $\text{ord}(2) = 2n$, así $(\frac{2n}{2n}, n) = (1, n) = 1$, por lo que se cumplen las condiciones para construir un período de Gauss.

(b) Si $< 2 >= RC(\mathbb{Z}^*_{2n+1})$, entonces el $\text{ord}(2) = n$, y como $2n + 1 \equiv 3 \bmod 4$, se cumple que n es impar y así $(\frac{2n}{n}, n) = (2, n) = 1$.

□

Si consideramos q primo, entonces cobstruir una base normal óptima de *Tipo I*, para $\mathbb{F}_{p^n}$ (ver sección 3.2.2), satisface las mismas condiciones dadas para construir una base normal generada por los períodos de Gauss con $k = 1$, $n+1$ primo, $(\frac{n}{\text{ord}(q)}, 1) = 1$, esto es, q es primitivo en $\mathbb{Z}^*_{n+1}$, con la salvedad que no podremos usar el teorema 3.3.9.

Ahora veremos cual es la complejidad de realizar la multiplicación, usando la representación de bases normales generadas por los períodos de Gauss, donde se aprovecha el hecho de que como β es una raíz $(nk+1)$-ésima de la unidad, se cumple $\sum_{i=0}^{nk} \beta^i = -1$.

Teorema 3.3.15. *Supongamos que representamos los elementos del campo finito $\mathbb{F}_{q^n}$ usando una base normal generada por un período de Gauss de tipo (n, k), entonces la multiplicación en $\mathbb{F}_{q^n}$ puede ser calculada con*

$$O(nklog(nk)loglog(nk))$$

operaciones sobre $\mathbb{F}_q$.

Demostración. Primero supongase que tenemos $\alpha, \beta, \alpha_i, \mathcal{K}_i$, para $0 \leq i \leq n-1$ como en el teorema 3.3.9. Entonces $\{\alpha_0, \alpha_1, \alpha_2, \ldots, \alpha_{n-1}\}$ es una base

normal para $\mathbb{F}_{q^n}$ sobre $\mathbb{F}_q$ generada por el período de Gauss del tipo (n,k). Sean

$$\gamma_1 = \sum_{i=0}^{n-1} a_i\alpha_i, \quad \gamma_2 = \sum_{i=0}^{n-1} b_i\alpha_i \quad \text{con } a_i, b_i \in \mathbb{F}_q,$$

elementos arbitrarios de $\mathbb{F}_{q^n}$. Queremos calcular $\gamma_1 \cdot \gamma_2 = \sum_{i=0}^{n-1} c_i\alpha_i$, con $c_i \in \mathbb{F}_q$. Escribimos γ_1 en términos de $1, \beta, \beta^2, \ldots, \beta^{nk}$:

$$\gamma_1 = \sum_{i=0}^{n-1} a_i \sum_{j\in\mathcal{K}_i} \beta^j = \sum_{j=1}^{nk} a'_j\beta^j,$$

con $a'_j = a_i$ si $j \in \mathcal{K}_i$. De manera análoga $\gamma_2 = \sum_{j=0}^{nk} b'_j\beta^j$. Los coeficientes $d_0, \ldots, d_{nk} \in \mathbb{F}_q$ de $\gamma_1\cdot\gamma_2 = \sum_{j=0}^{nk} d_j\beta^j$ pueden ser calculados usando multiplicación rápida de polinomios en $O(nklog(nk)loglog(nk))$ (ver [CaKa91, Sc77]) operaciones sobre $\mathbb{F}_q$. También notamos que $\gamma_1 \cdot \gamma_2 = \sum_{j=0}^{nk} c'_j\beta^j$ donde $c'_j = d_j - d_0$, puesto que $\sum_{j=1}^{n-1} \beta^j = -1$. Para $0 \le i \le n-1$, sea $c_i = c'_j$ para $j \in \mathcal{K}_i$, notando que c'_j es el mismo para toda $j \in \mathcal{K}_i$. Entonces $\gamma_1 \cdot \gamma_2 = \sum_{i=0}^{n-1} c_i\alpha_i$. Con esto se demuestra que $\gamma_1 \cdot \gamma_2$ puede ser calculado en $O(nklog(nk)loglog(nk))$ operaciones en $\mathbb{F}_q$.

□

Si consideramos k constante, entonces la complejidad de la multiplicación es $O(n\,logn\,loglogn)$.

3.3.3 Exponenciación

En [GaGaPa95, GaGaPa98] se presenta un algoritmo que realiza la exponenciación, primero se considera la de un período de Gauss la cual se realiza en $O(n^2)$ operaciones sobre $\mathbb{F}_q$ y la de un elemento cualquiera requiere $O(n^2loglogn)$.

Teorema 3.3.16. *Sea α un período de Gauss del tipo (n,k) sobre $\mathbb{F}_q$ y $0 \le M \le q^n - 1$. Entonces α^M puede ser calculado con $O(n^2qk)$ operaciones sobre $\mathbb{F}_q$.*

Demostración. Deseamos calcular α^M expresado en términos de la base $BNPG$

$$\{\alpha_0, \alpha_1, \ldots, \alpha_{n-1}\}.$$

Para esto usamos una representación redundante de $\gamma \in \mathbb{F}_{q^n}$, escribiendo

$$\gamma = \left(\sum_{i=0}^{n-1} a_i \alpha_i \right) + a_n, \quad con\ a_i \in \mathbb{F}_q.$$

Así γ esta representada por un arreglo de $(n+1)$ entradas, es decir,

$$(a_0, a_1, \ldots, a_{n-1}, a_n).$$

Claramente, esta representación no es única, en este caso se usará cualquiera. Por ejemplo, el 1 esta representado por $(0, \ldots, 0, 1)$ o $(-1, \ldots, -1, 0)$ recordando que la $Tr(\alpha_0) = -1$.

Cuando se calcula γ^q, su representación es el corrimiento de las primeras n posiciones a la derecha y la posición $n+1$ permanece fija. Puesto que

$$\begin{aligned}(\gamma)^q &= \left(\left(\sum_{i=0}^{n-1} a_i \alpha_i \right) + a_n \right)^q \\ &= \left(\sum_{i=0}^{n-1} a_i \alpha_i^q \right) + a_n \quad \text{por el lema 1.2.4.}\end{aligned}$$

Esta operación es computacionalmente insignificante.

Para cualquier $\gamma \in \mathbb{F}_{q^n}$ y $0 \leq j \leq n-1$,

$$\alpha_j \gamma = \left(\sum_{i=0}^{n-1} a_i (\alpha \alpha_{i-j})^{q^j} \right) + a_n \alpha_j.$$

Por el lema 3.3.8 el producto $\alpha\alpha_i$ tiene a lo más k términos. Por lo tanto el producto $\alpha_j \gamma$ puede ser calculado con $O(nk)$ operaciones sobre $\mathbb{F}_q$, ya que se hacen n sumas de k elementos.

Ahora, para calcular α^M, usamos la representación q-aria de

$$M = \sum_{j=0}^{l} m_j q^j,$$

con $0 \leq m_j \leq q-1$ para toda j y $m_l \neq 0$. Entonces $l < n$, y notemos que

$$\alpha^M = \prod_{j=0}^{l} (\alpha^{q^j})^{m_j} = \prod_{j=0}^{l} \alpha_j^{m_j}.$$

Por lo que calcularemos α^M usando el siguiente algoritmo:

Algoritmo 3.3.17.

1. $\gamma := 1$.
2. *Calculamos para* $0 \leq j \leq l$

$$\gamma := \alpha_j^{M_j}\gamma.$$

3. *finalmente, obtenemos* $\gamma = \alpha^M$

Este algoritmo calcula α^M en

$$O\left(\left(\sum_{j=0}^{l} M_j\right) nk)\right) = O(w_q(M)nk)$$

operaciones sobre $\mathbb{F}_q$, donde, $w_q(M)$, es la suma de los dígitos de M en la representación q-aria. Entonces como $w_q(M) \leq (q-1)n < qn$ así obtenemos

$$O(qn^2k).$$

□

Si los valores k y q son fijos, entonces α^M se calcula en $O(n^2)$ operaciones.

Para calcular la exponenciación de un elemento cualquiera $\gamma = \sum_{i=0}^{n-1} a_i\alpha_i + a_n$, digamos γ^M, volvemos a considerar la expansión q-aria de M y el hecho que:

$$\gamma^M = \prod_{j=0}^{n-1}(\gamma^{q^j})^{M_j} = \prod_{j=0}^{n-1}((\sum_{i=0}^{h-1} a_i\alpha_i)^{q^j})^{M_j} = \prod_{j=0}^{n-1}(\sum_{i=0}^{h-1} a_i\alpha_i^{q^j})^{M_j},$$

donde $0 \leq M_j \leq q-1$ llevará $O(n^2 lognlogn)$ operaciones sobre $\mathbb{F}_q$, puesto que se realizan más multiplicaciones.

En la tabla 3.2 (fragmento de [GaGaPa95]) podemos observar parejas de enteros que satisfacen que $nk+1$ es primo. Estas parejas no estan relacionados con ninguna $q = p^m$ para alguna m.

(2,2)	(2,5)	(2,6)	(2,9)	(2,14)	(2,18)	(3,2)	(3,4)
(3,6)	(3,12)	(3,20)	(4,3)	(4,7)	(4,9)	(5,2)	(5,6)
(5,8)	(6,2)	(6,3)	(7,4)	(7,6)	(9,2)	(9,4)	(9,8)
(10,6)	(11,2)	(11,6)	(11,8)	(12,3)	(12,5)	(13,4)	(13,6)
(13,10)	(14,4)	(14,12)	(17,6)	(17,8)	(17,14)	(18,2)	(18,9)
(18,10)	(19,10)	(19,12)	(20,3)	(20,5)	(20,9)	(21,10)	(21,16)
(21,18)	(22,3)	(22,18)	(23,2)	(23,6)	(23,12)	(25,4)	(25,16)
(26,2)	(26,5)	(26,6)	(27,6)	(27,10)	(27,14)	(28,7)	(28,15)
(29,2)	(29,8)	(29,12)	(30,2)	(30,6)	(30,7)	(30,11)	(30,14)
(31,10)	(31,12)	(33,2)	(33,6)	(33,14)	(34,9)	(34,13)	(35,2)
(35,6)	(35,8)	(35,12)	(35,14)	(36,5)	(36,15)	(36,17)	(37,4)
(37,6)	(37,16)	(38,6)	(38,11)	(38,15)	(39,2)	(39,8)	(39,14)
(41,2)	(41,18)	(41,20)	(42,5)	(42,9)	(42,10)	(43,4)	(43,10)
(44,9)	(44,15)	(45,4)	(45,6)	(45,12)	(45,14)	(46,3)	(46,6)
(46,10)	(46,15)	(47,6)	(47,14)	(47,20)	(49,4)	(49,10)	(49,18)
(50,2)	(50,14)	(51,2)	(51,8)	(51,12)	(52,3)	(52,13)	(53,2)
(53,14)	(53,20)	(54,3)	(54,7)	(54,10)	(54,14)	(55,12)	(55,16)
(55,18)	(57,10)	(58,6)	(58,9)	(59,12)	(59,14)	(59,18)	(60,3)
(60,7)	(60,9)	(60,11)	(61,6)	(61,12)	(61,16)	(62,6)	(63,6)
(65,2)	(66,10)	(67,4)	(68,9)	(69,2)	(70,3)	(71,8)	(73,4)
(74,2)	(75,10)	(76,3)	(77,6)	(78,7)	(79,4)	(81,2)	(82,9)
(83,2)	(84,5)	(85,12)	(86,2)	(87,4)	(89,2)	(90,2)	(91,6)
(92,3)	(93,4)	(94,3)	(95,2)	(97,4)	(98,2)	(99,2)	(100,7)
(101,6)	(102,6)	(103,6)	(105,2)	(103,10)	(107,6)	(108,5)	(109,10)
(110,6)	(111,20)	(113,2)	(114,5)	(115,4)	(116,3)	(117,8)	(118,6)
(119,2)	(121,6)	(122,6)	(123,10)	(124,3)	(125,6)	(126,3)	(127,4)
(129,8)	(130,9)	(131,2)	(132,5)	(133,2)	(134,2)	(135,2)	(137,6)
(138,6)	(139,4)	(140,3)	(141,8)	(142,6)	(143,6)	(145,10)	(146,2)
(147,6)	(148,15)	(149,8)	(150,19)	(151,6)	(153,4)	(154,25)	(155,2)
(156,13)	(157,10)	(158,2)	(159,22)	(161,6)	(162,10)	(163,4)	(164,5)
(165,4)	(166,3)	(167,14)	(169,4)	(170,6)	(171,12)	(172,9)	(173,2)
(174,2)	(175,4)	(177,4)	(178,6)	(179,2)	(180,3)	(181,6)	(182,3)
(183,2)	(185,8)	(186,2)	(187,6)	(188,5)	(189,2)	(190,10)	(191,2)
(193,4)	(194,2)	(195,6)	(196,7)	(197,18)	(198,22)	(199,4)	(201,8)
(202,6)	(203,12)	(204,3)	(205,4)	(206,3)	(207,4)	(209,2)	(210,2)
(211,10)	(212,5)	(213,4)	(214,5)	(215,6)	(217,6)	(218,5)	(219,4)

Tabla 3.2: Tabla de pares (n, k) (con $n < 220$) de Gauss

Capítulo 4

Aplicaciones Criptográficas

El objetivo de este capítulo es presentar algunos criptosistemas que requieren hacer operaciones sobre campos finitos. Puesto que cuando se implementan estos métodos de cifrado se necesita obtener la información cifrada en el menor tiempo posible, es decir, que se haga en forma muy rápida, esto no sólo depende del diseño de la implementación, sino que también se necesita que las operaciones básicas como suma, multiplicación y exponenciación se realicen en forma eficiente, es decir, en poco tiempo.

Encontramos que hay criptosistemas tanto de llave privada como de llave pública que realizan operaciones sobre campos finitos. Por ejemplo, en criptografía de llave pública, tenemos intercambio de llaves Diffie-Hellman [DiHe76], criptosistema ElGamal [El85] y criptosistemas basados en la aritmética de puntos racionales de Curvas Elípticas [Ko87, Mi86]. En criptografía simétrica el sistema criptográfico que reemplazará al DES [FIPS46], el Rijndael [DaRi98], sus transformaciones realizan operaciones sobre $\mathbb{F}_{2^8}$.

En herramientas usadas en criptografía como lo es la generación de números pseuodoaleatorios [BlMi84], también se requiere efectuar exponenciación eficiente sobre campos finitos. En este capítulo se describirán brevemente el criptosistema Rijndael, el intercambio de llaves Diffie-Hellman y un criptosistema basado en Curvas Elípticas. Para mayor información ver las referencias [DiHe76, DaRi98, Ko87, Mi86].

4.1 AES Rijndael

En 1996 el National Institute of Standards and Technology (NIST[1]) inició un programa para elegir un Advanced Encryption Standard (AES) para reemplazar al DES, anunciando en octubre del 2000 como ganador al algoritmo Rijndael, diseñado por los belgas J. Daemen y V. Rijmen [DaRi98].

Rijndael es un cifrado en bloque cuya entrada pueden ser bloques de tamaño variable, y cuya llave también es de tamaño variable, el bloque y la llave pueden ser de 128, 192 o 256 bits.

Las transformaciones que realiza son a nivel de bytes[2]. Efectuando esta aritmética usando bases polinomiales[3] del campo finito $\mathbb{F}_{2^8}$ sobre $\mathbb{F}_2$, definidas en términos de un polinomio irreducible $m(x)$ de grado 8 sobre $\mathbb{F}_2$.

Por ejemplo, usando el polinomio irreducible $p(x) = x^8 + x^4 + x^3 + x + 1$ para definir el campo finito:

$$\mathbb{F}_{2^8} = \{\sum_{i=0}^{7} a_i \alpha^i : \quad \text{con } a_i \in \mathbb{F}_2 \quad \text{y} \alpha^8 = \alpha^4 + \alpha^3 + \alpha + 1\}.$$

Claramente una base polinomial es: $\{1, \alpha, \ldots, \alpha^7\}$. Ya que se eligió la base, veremos que se hacen multiplicaciones y cálculo de inversos (es decir, exponenciación) en las transformaciones que describiremos a continuación.

Definición 4.1.1. *Sean* $Nb = \dfrac{|bloque|}{32}$ *y* $Nk = \dfrac{|llave|}{32}$.

El bloque de bits a cifrar y la llave son vistos como matrices de bytes $b_{4\times Nb}$ y $k_{4\times Nk}$ respectivamente. Nb y Nk son el número de columnas de matrices de 4 renglones que se calculan como en la definición 4.1.1, esto es, Nb y Nk toman los valores 4 si son de 128 bits, de 6 si son de 192 bits y de 8 si son de 256 bits.

Por ejemplo, de un bloque con 128 bits 01010000 10001000 10001100 10010011 10100101 10101011 01100001 11100101 01100111 00010111 10011011

[1] Instituto norteamericano encargado de la administración de los Estándares de Tecnología.

[2] Estos elementos a lo largo de este trabajo son representados con notación hexadecimal.

[3] Puesto que los autores consideran que estas son más adecuadas que las bases normales.

10011110 01001001 01010110 11110111 01110111 obtenemos la matriz $b_{4\times 4}$ de bytes (tomando cada 8 bits y escribiendo el número hexadecimal correspondiente, por ejemplo, 01010000=50), es decir, con $Nb = 4$ columnas.

$b=$

50	88	8C	93
A5	AB	61	E5
67	17	9B	9E
49	56	F7	77

El número de iteraciones que realiza el cifrado Rijndael es denotado por Nr. Este depende de Nb y Nk y esta dado por la tabla 4.1.

Nr	$Nb = 4$	$Nb = 6$	$Nb = 8$
$Nk = 4$	10	12	14
$Nk = 6$	12	12	14
$Nk = 8$	14	14	14

Tabla 4.1: Nr número de iteraciones en función de Nb y Nk

Las transformaciones que efectua Rijndael a el bloque son ByteSub, ShiftRow, MixColumn y AddRoundKey.

4.1.1 Transformación ByteSub

Esta transformación se aplica a cada byte de la matriz $b_{4\times Nb}$, y consiste de la composición de dos funciones.

$$f_1(x) = \begin{cases} x^{-1}, & \text{si } x \neq '00' \\ '00', & \text{si } x = '00'. \end{cases} \tag{4.1}$$

y la función

$$f_2(x) = Sx \oplus v \tag{4.2}$$

con $x \in \mathbb{F}_{2^8}$ en la ecuación (4.2). S es una matriz de bits, esta es fija e invertible de tamaño 8×8 por ejemplo:

$$S = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

y v también es un byte fijo, por ejemplo:

$$v = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

ambos son seleccionados cuidadosamente para que sea difícil aplicar el criptoanálisis diferencial (ver [Ny94]).

Finalmente la transformación ByteSub esta dada por

$$BS(b_{ij}) = f_2(f_1(b_{ij})) \tag{4.3}$$

$$f_2^{-1}(x) = S^{-1}(x \oplus v) \tag{4.4}$$

donde $b_{ij} \in \mathbb{F}_{2^8}$. Esta función da como resultado otro byte. La inversa de esta transformación es la composición de las ecuaciones (4.4) y (4.1).

Así, la inversa de ByteSub esta dada por

$$BS^{-1}(b_{ij}) = f_2^{-1}(f_1(b_{ij})). \tag{4.5}$$

En esta transformación podemos notar la importancia de calcular inversos en forma eficiente. Encontrar el inverso de $x \in \mathbb{F}_{2^8}$ en este caso es calcular $x^{-1} = x^{2^8-2}$, es decir, efectuar una exponenciación. Estas operaciones dependen de la base escogida, puesto que con las propiedades de la base propuesta se efecturan mas rápido.

Nb	$C1$	$C2$	$C3$
4	1	2	3
6	1	2	3
8	1	3	4

225993

Tabla 4.2: Corrimientos de la función ShiftRow

4.1.2 Transformación ShiftRow

La transformación ShiftRow consiste de corrimientos de los renglones del bloque de acuerdo con la tabla 4.2, sólo el renglón R_0 no sufre corrimiento.

La función $SR : \mathbb{F}_{2^8} \mapsto \mathbb{F}_{2^8}$ que define la transformación ShiftRow esta dada por la ecuación (4.6), que usa los datos de la tabla 4.2.

$$SR(b_{i,j}) = \begin{cases} b_{i,j+C1 \bmod Nb}, & \text{si } i = 1 \\ b_{i,j+C2 \bmod Nb}, & \text{si } i = 2 \\ b_{i,j+C3 \bmod Nb}, & \text{si } i = 3. \end{cases} \tag{4.6}$$

con $0 \leq j \leq Nb - 1$.

Esta transformación no depende de la aritmética sobre el campo finito, puesto que sólo consiste de corrimientos.

4.1.3 Transformación MixColumn

La función

$$MC : \mathbb{F}_{2^8}[x]/ < x^4 + 1 > \longmapsto \mathbb{F}_{2^8}[x]/ < x^4 + 1 >$$

($\mathbb{F}_{2^8}[x]/ < x^4 + 1 > \simeq \mathbb{F}_{2^8}^4$ como espacio vectorial, podemos ver a los polinomios como vectores de longitud 4) define la transformación MixColumn:

$$MC(b_0, b_1, b_2, b_3) = \begin{bmatrix} c_0 & c_3 & c_2 & c_1 \\ c_1 & c_0 & c_3 & c_2 \\ c_2 & c_1 & c_0 & c_3 \\ c_3 & c_2 & c_1 & c_0 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} \tag{4.7}$$

donde (b_0, b_1, b_2, b_3) es una columna del bloque b, donde $b_i, c_i \in \mathbb{F}_{2^8}$ y

$$c(x) = c_3x^3 + c_2x^2 + c_1x + c_0 \in \mathbb{F}_{2^8}[x]/ < x^4 + 1 >$$

es un polinomio fijo de grado 4 primo relativo a x^4+1. Por lo tanto la matriz circulante formada por los coeficientes de $c(x)$, es invertible.

La inversa de la transformación MixColumn usa la inversa de la matriz formada por los coeficientes del polinomio $c(x)$. El cálculo de esta inversa requiere más operaciones sobre el campo finito $\mathbb{F}_{2^8}$.

En este caso se requiere hacer la multiplicación de una matriz de 4×4 por un vector de longitud 4. Entonces una evaluación de esta transformación requiere 16 multiplicaiones y 16 sumas sobre $\mathbb{F}_{2^8}$. Por esta razón es conveniente que se hagan en forma rápida estas operaciones.

4.1.4 Transformación AddRoundAddition

Esta transformación consiste simplemente de aplicar una vez la operación lógica $EXOR$ del bloque con la subllave k_i de la iteración correspondiente.

$b_{0,0}$	$b_{0,1}$	$b_{0,2}$	$b_{0,3}$	$b_{0,4}$	$b_{0,5}$
$b_{1,0}$	$b_{1,1}$	$b_{1,2}$	$b_{1,3}$	$b_{1,4}$	$b_{1,5}$
$b_{2,0}$	$b_{2,1}$	$b_{2,2}$	$b_{2,3}$	$b_{2,4}$	$b_{2,5}$
$b_{3,0}$	$b_{3,1}$	$b_{3,2}$	$b_{3,3}$	$b_{3,4}$	$b_{3,5}$

$\oplus$

$k_{0,0}$	$k_{0,1}$	$k_{0,2}$	$k_{0,3}$	$k_{0,4}$	$k_{0,5}$
$k_{1,0}$	$k_{1,1}$	$k_{1,2}$	$k_{1,3}$	$k_{1,4}$	$k_{1,5}$
$k_{2,0}$	$k_{2,1}$	$k_{2,2}$	$k_{2,3}$	$k_{2,4}$	$k_{2,5}$
$k_{3,0}$	$k_{3,1}$	$k_{3,2}$	$k_{3,3}$	$k_{3,4}$	$k_{3,5}$

la inversa de esta transformación es ella misma.

4.1.5 Expansión de la llave en subllaves

Usando la llave k construimos subllaves k_i formadas por palabras de 4 bytes, $W_0, W_1, \ldots, W_{Nb(Nr+1)}$. El número de palabras necesario para construir las subllaves es $Nb(Nr+1)$, ya que se necesita una subllave formada por Nb palabras W_i para cada iteración. Estas palabras se forman con funciones que modifican la llave usando la operación lógica $EXOR$ y corrimientos (ver [DaRi98]).

Una iteración de Rijndael consiste de aplicar al bloque, la transformaciones en el siguiente orden:

1. ByteSub
2. ShiftRow
3. MixColumn
4. aplicar AddRoundKey usando la subllave correspondiente de la iteración.

Además de una iteración final que consiste de aplicar lo anterior excepto MixColumn.

Una ventaja de utilizar Rijndael, es que como entrada puede tener distintos tamaños de bloques así como de llaves, lo cual da una gama de combinaciones, otra es que no depende específicamente de las S-cajas. Una desventaja puede ser que las operaciones de descifrado, es decir, las transformaciones inversas son un poco mas complicadas que el cifrado, además de ser distintas[4].

Para mejorar la implementación del cifrado Rijndael, se deben encontrar mejores métodos para efectuar multiplicaciones y exponenciaciones sobre $\mathbb{F}_{2^8}$. Es decir, debemos encontrar la base que facilite la aritmética, ya que como hemos visto todas las transformaciones descritas dependen de hacer esta aritmética rápidamente.

4.2 Intercambio de llaves

Este método fue propuesto por W. Diffie y M. E. Hellman [DiHe76] a mediados de los años 70's. Cuando dos usuarios desean hacer este tipo de intercambio seleccionan primero un campo finito $\mathbb{F}_{q^n}$ y un generador g de este. Si el usuario A decide acordar una llave con el usuario B, ambos escogen enteros $2 < x_A, x_B < |(\mathbb{F}_{q^n})^* - 1$ respectivamente. A envia al usuario B el elemento del campo finito que resulta de elevar el generador al número que escogió

$$K_A = g^{x_A},$$

entonces B hace lo mismo enviando

$$K_B = g^{x_B}.$$

[4]Esto es que como se tienen que calcular inversas además de otras operaciones que dependen del polinomio irreducible elegido para realizar las operaciones.

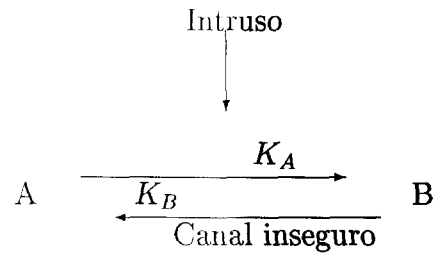

Figura 4.1: Esquema de intercambio de llaves Diffie-Hellman

Al recibir K_A y K_B respectivamente, cada uno eleva el número que eligió al número que ha recibido.

$$K_A^{x_B} = (g^{x_A})^{x_B} = g^{x_A x_B} = K$$

y

$$K_B^{x_A} = (g^{x_B})^{x_A} = g^{x_B x_A} = K$$

De esta forma cada usuario obtiene la misma llave, que pueden utilizar con un algoritmo de llave secreta. El esquema de intercambio se ve en la figura 4.1.

El intruso que pudiera obtener K_A o K_B se enfrenta al problema de logaritmo discreto, ya que tendría que encontrar x_A o x_B. Por esta razón este tipo de intercambio es seguro si se utiliza un campo finito con orden grande ya que este es público.

Con el avance de las computadoras el campo finito $\mathbb{F}_{q^n}$ que se usa en este proceso es más grande. Entonces para implementar este esquema de intercambio de llaves los usuarios necesitan tener métodos eficientes de exponenciación sobre campos finitos y esto esta directamente ligado a la base utilizada para representar a los elementos del campo finito.

4.3 Criptosistema de Curvas Elípticas

El criptosistema de curvas elípticas fue propuesto simultáneamente por N. Koblitz [Ko87] y V. Miller [Mi86] a mediados de los años 80's.

Este criptosistema se basa en el hecho de que el conjunto de puntos racionales[5] de una curva elíptica, es decir, el conjunto de soluciones de una ecuación del tipo (4.8), es un grupo conmutativo.

$$y^2 = x^3 + ax + b \tag{4.8}$$

En este trabajo no mencionamos los cambios de variable necesarios para llevar de la fórmula general a cada una de las curvas que veremos a continuación, como referencias tenemos [Me93, Si86].

Definición 4.3.1 (Curva Elíptica). *Una curva elíptica sobre un campo* $\mathbf{K}$ *es el conjunto de pares ordenados* $(x, y) \in \mathbf{K} \times \mathbf{K}$, *que son solución a la ecuación (4.8) y cumple que* $4a^3 + 27b^2 \neq 0$ *o que* $x^3 + ax + b$ *no tiene factores repetidos [Ko87] y el punto* $\mathcal{O}$ *al infinito.*

4.3.1 Curvas Elípticas sobre los reales

Primero explicaremos como se hace la "suma" de puntos racionales de una curva elíptica sobre los números reales $\mathbb{R}$, recordando que la $x^3 + ax + b$ no debe tener factores repetidos.

Ejemplo 4.3.2. *Si la ecuación es (4.8) la gráfica (real) de la curva elíptica es como en la figura 4.2.*

El conjunto

$$E = \{(x, y) : y^2 = x^3 + ax + b \quad x, y, a, b \in \mathbb{R}\} \cup \mathcal{O}$$

forma un grupo abeliano, donde $\mathcal{O}$ es el punto al infinito. La "suma"de $P, Q \in E$ esta definida por la recta que pasa por los puntos P, Q, siendo el resultado de la suma el punto del grupo E que es punto de reflexión de la intersección de la recta $\overline{PQ}$ con la curva E.

Veremos esta suma geométricamente.

1. Si $P, Q \in E$ son dos puntos distintos la suma se define como en la figura 4.3.
2. El doble de un punto $P = (x_P, y_P) \in E$ tiene dos casos
   (a) Si $y_P \neq 0$ entonces la suma $P + P = 2P = R$ esta dada por la figura 4.4.

[5]Nos referiremos a los puntos $\mathbb{F}_{q^n}$-racionales sólo como puntos racionales.

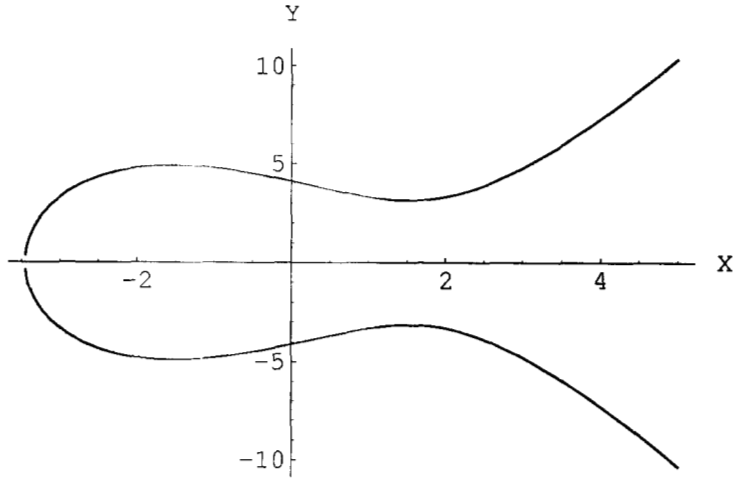

Figura 4.2: Curva Elíptica $y^2 = x^3 - 7x + 17$

(b) Si $y_P = 0$ el doble del punto P se define en la figura 4.5

3. El elemento identidad es el punto al infinito $\mathcal{O}$.

4. El inverso de un punto $P = (x_P, y_P)$ es $-P = (x_P, -y_P)$ (que también pertenece a la curva). La suma de los puntos P, $-P$ se ve en la figura 4.6.

Analíticamente la suma esta dada de la siguiente forma:

1. Cuando los puntos $P = (x_P, y_P)$, $Q = (x_Q, y_Q)$ son distintos. Calculamos la recta que pasa por P y Q.

   La pendiente corresponde a la ecuación (4.9), y la recta a la ecuación (4.10)

$$m = \frac{y_P - y_Q}{x_P - x_Q} \tag{4.9}$$

$$y = m(x - x_P) + y_P \tag{4.10}$$

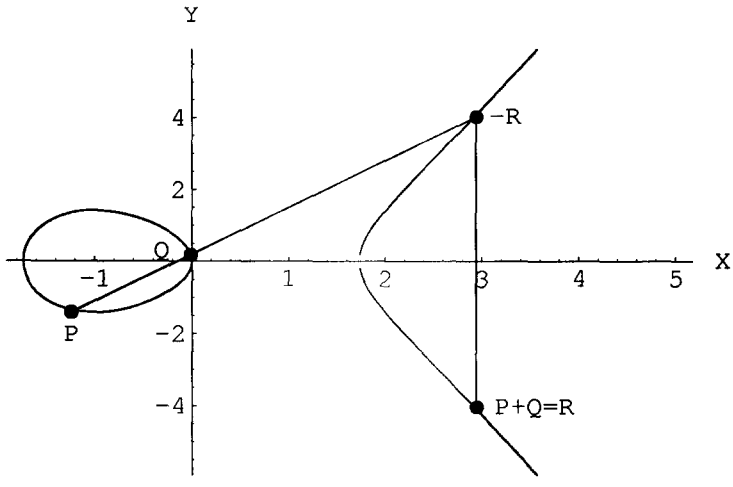

Figura 4.3: $P + Q = R$

El punto $-R$ es la intersección de la recta (4.10) con la curva (4.8). Al sustituir la recta (4.10) en la curva (4.8) obtenemos la ecuación (4.11)

$$(m(x - x_P) + y_P)^2 = x^3 + ax + b \tag{4.11}$$

Resolviendo la ecuación de tercer grado (4.11) obtenemos

$$-R = (x_R, -y_R) = (m^2 - x_P - x_Q, y_P + m(x_R - x_P)) \tag{4.12}$$

Finalmente de la ecuación (4.12) obtenemos la reflexión de $-R$ obteniendo el punto R.

$$R = (x_R, y_R) = (m^2 - x_P - x_Q, -y_P + m(x_P - x_R)) \tag{4.13}$$

2. Para calcular el doble de un punto P cuando $y_P \neq 0$. Calculamos la recta tangente que pasa por P, haciendo la sustitución como en la suma de dos puntos distintos obtenemos el punto R.

$$m = \frac{3{x_P}^2 + a}{2y_P} \tag{4.14}$$

$$R = (x_R, y_R) = (m^2 - 2x_P, -y_P + m(x_P - x_R)) \tag{4.15}$$

donde a corresponde al coeficiente lineal en la curva elíptica (4.8)

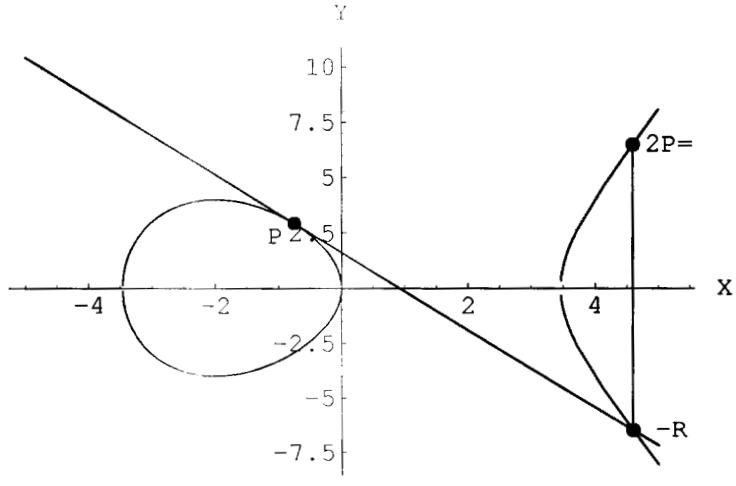

Figura 4.4: $2P = R$

4.3.2 Curvas Elípticas sobre campos finitos $\mathbb{F}_{q^n}$

En el caso de los campos finitos $\mathbb{F}_{q^n}$ con característica $\neq 2$, 3, el conjunto

$$E = \{(x, y) : y^2 = x^3 + ax + b \quad a, b, x, y \in \mathbb{F}_{q^n}\} \cup \mathcal{O}$$

esta formado por un número finito de parejas de acuerdo con el Teorema de Hasse.

Teorema 4.3.3 (Hasse). *Sea la curva E como en la definición 4.3.1 con $K = \mathbb{F}_{q^n}$. El número N de puntos racionales de E sobre $\mathbb{F}_{q^n}$ satisface*

$$|N - (q^n + 1)| \leq 2\sqrt{q^n}.$$

Demostración. Ver [Ca91] Teorema 1 página 118. □

La suma esta definida de la misma forma que en los reales, claro que realizando las operaciones suma, resta, multiplicación y división del campo finito de acuerdo a la base escogida.

Para campos de caraterística 2, no se usa la misma representación de la curva (4.8), sino que, se pueden usar las curvas no equivalentes (4.16) y (4.17).

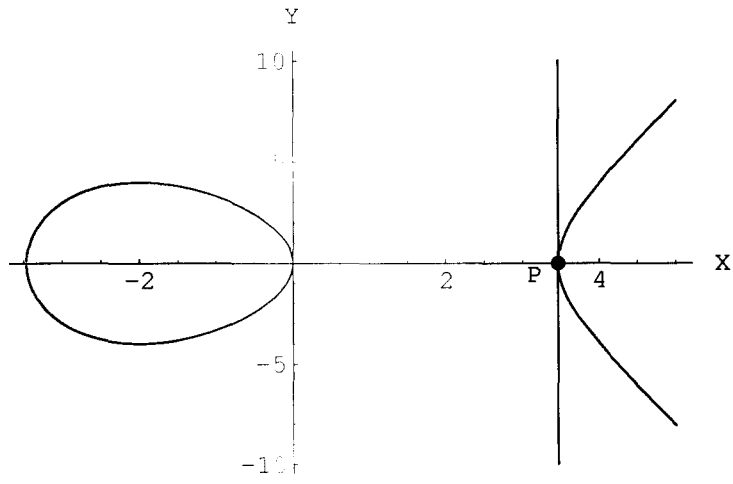

Figura 4.5: $2P = \mathcal{O}$

$$y^2 + xy = x^3 + ax^2 + b \tag{4.16}$$

$$y^2 + cy = x^3 + ax + b \tag{4.17}$$

Con estas dos ecuaciones definimos a continuación la aritmética correspondiente.

El conjunto

$$E = \{(x, y) : y^2 + xy = x^3 + ax^2 + b \quad x, y, a, b \in \mathbb{F}_{2^n}\} \cup \mathcal{O}$$

forma un grupo abeliano, donde $\mathcal{O}$ es el punto al infinito.

La operación suma para el conjunto E se define como sigue:

1. Si los puntos $P = (x_P, y_P)$, $Q = (x_Q, y_Q) \in E$ son distintos y $P \neq -Q$, la suma $P + Q = R$ donde R es como en la ecuación (4.19).

$$m = \frac{y_P + y_Q}{x_P + x_Q} \tag{4.18}$$

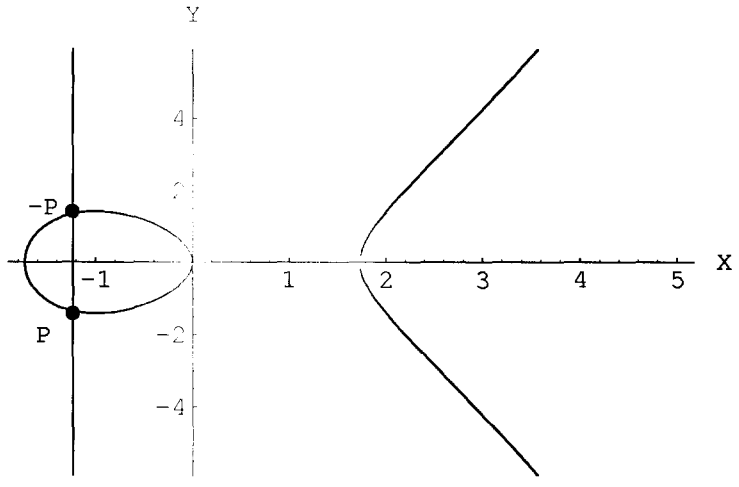

Figura 4.6: $P - P = \mathcal{O}$

$$R = (x_R, y_R) = (m^2 + m + x_P + x_Q + a, m(x_R + x_P) + y_P + x_R)) \tag{4.19}$$

El cálculo de esta pendiente requiere 2 sumas, el cálculo de un inverso (que involucra exponenciación) y una multiplicación sobre el campo finito usado.

2. Si $x_P = 0$ entonces $2P = \mathcal{O}$, si $x_P \neq 0$,

$$m = \frac{{x_P}^2 + y_P}{x_P}$$

$$2P = R = (x_R, y_R) = (m^2 + m + a, x_P^2 + x_R(m+1))$$

donde a es el coeficiente cuadrático de la curva (4.16). En este caso se requiere realizar una suma, una exponenciación, una multiplicación y el cálculo de un inverso (es decir, exponenciación).

3. El neutro es el punto al infinito $\mathcal{O}$.

4. El inverso de $P = (x_P, y_P)$ es $-P = (x_P, x_P + y_P)$ que también es punto de la curva (4.16), la suma $P + (-P) = \mathcal{O}$.

Ahora el conjunto definido por la curva (4.17)

$$E = \{(x, y) : y^2 + cy = x^3 + ax + b \quad x, y, a, b, c \in \mathbb{F}_{2^n}\} \cup \mathcal{O}$$

donde $\mathcal{O}$ es el punto al infinito, forma también un grupo abeliano.

La operación suma para este nuevo conjunto E se define:

1. Si los puntos $P = (x_P, y_P)$, $Q = (x_Q, y_Q) \in E$ son distintos y $P \neq -Q$, la suma $P + Q = R$ donde R es como en la ecuación (4.21).

$$m = \frac{y_P + y_Q}{x_P + x_Q} \tag{4.20}$$

$$R = (x_R, y_R) = (m^2 + x_P + x_Q, m(x_R + x_P) + y_P + c)) \tag{4.21}$$

   El cálculo del punto R requiere para la primera entrada 1 exponenciación y 3 sumas, la segunda entrada una multiplicación y 4 sumas.

2. Si $x_P = 0$ entonces $2P = \mathcal{O}$, si $x_P \neq 0$,

$$m = {x_P}^2 + a$$

$$2P = R = (x_R, y_R) = (m^2, m(x_P + x_R) + y_p + c))$$

   donde c es el coeficiente lineal de y de la curva (4.17).

3. El neutro es el punto al infinito $\mathcal{O}$.

4. El inverso de $P = (x_P, y_P)$ es $-P = (x_P, x_P + c)$ que también es punto de la curva (4.17), la suma $P + (-P) = \mathcal{O}$.

En este trabajo no vemos el caso de la aritmética de curvas definidas sobre un campo de característica 3. Como vemos efectuar la aritmética de puntos racionales sobre curvas elípticas en forma rápida, depende de la elección de la base y del desarrollo de métodos de multiplicación y exponenciación sobre campos finitos. No consideramos la división puesto que el calculo de inversos sobre un campo finito es finalmente una exponenciación. Además podemos observar que las operaciones sobre campos finitos de característica 2 son mas sencillas.

4.3.3 Criptosistemas usando curvas elípticas

El grupo E definido en la sección 4.3.2 sobre curvas elípticas puede ser usado en el esquema de intercambio de llaves Diffie-Hellman [DiHe76] y el criptosistema ElGamal [El85]. Puesto que el problema de logaritmo discreto sobre el grupo E, es difícil de resolver.

Versión elíptica del Problema de Logaritmo Discreto (PLD)

Sea E una curva elíptica definida sobre un campo finito $\mathbb{F}_{q^n}$. Sea $P \in E$ donde P es un punto racional cuyo orden d es muy grande. Sea $Q \in < P >$ esto es existe $r \in \mathbb{Z}$ tal que $Q = rP$, donde, $1 < r < d - 1$. El problema de logaritmo discreto consiste en encontrar r conociendo la curva elíptica E el punto P y Q. Encontrar el entero r es computacionalmente imposible si el orden d del punto base P es de aproximadamente 512 bits.

Versión elíptica del esquema de intercambio de llaves Diffie-Hellman

Si las personas A y B desean usar curvas elípticas para acordar una llave deben hacer lo siguiente:

1. A y B eligen una curva elíptica sobre un campo finito. Sea E el grupo definido por esta curva y el punto al infinito. Además escogen un punto $P \in E$ con orden d lo suficientemente grande.

2. Cada uno escoge un entero r_A y r_B repectivamente, donde, $1 < r_A, r_B < d - 1$.

3. A envia $Q_A = r_A P$ a B. B envia $Q_B = r_B P$ a A.

4. Cada uno obtiene por su parte $Q_{AB} = r_A r_B P$ al calcular $r_A Q_B$ y $r_B Q_A$ respectivamente, puesto que $r_A r_B = r_B r_A$.

Si un alguien quisiera conocer Q_{AB}, tendría que resolver el problema de logaritmo discreto encontrando r_A o r_B.

Versión elíptica del cifrado ElGamal

El cifrado ElGamal [El85] fue propuesto inicialmente para campos finitos. Ahora veremos el procedimiento usando curvas elípticas.

Sea E el grupo definido por una curva elíptica. De nuevo sea P el punto base de orden d.

Los pasos a seguir para cifrar un mensaje M son:

1. El usuario A escoge un entero a, donde, $1 < a < d - 1$. Calcula $Q_A = aP$.
2. A hace públicos E, P y Q_A, manteniendo en secreto al entero a.
3. Si el usuario B quiere enviarle un mensaje M a A. Primero asocia un punto racional P_M al mensaje M (o parte del mensaje).
4. B escoge al azar un entero k, donde, $1 < k < d - 1$.
5. B calcula los puntos $Q_1 = kP$ y $Q_2 = P_M + kQ_A$ y se los envia a A.
6. A al recibir el par de puntos (Q_1, Q_2) recupera el mensaje al calcular usando su entero secreto a lo siguiente:

$$Q_2 - aQ_1 = (P_M + kQ_A) - a(kP) = P_M + k(aP) - a(kP) = P_M.$$

Como podemos observar la seguridad de este cifrado se basa en la dificultad de encontrar el entero k, aun conociendo los puntos P y Q_1. Notamos además que tampoco A conoce al entero k puesto que sólo recibe al punto Q_1. También notemos que como se debe usar un campo finito grande para hacer más difícil la busqueda exhaustiva de la llave, con esto la implementación recae en considerar una base adecuada para realizar la aritmética eficiente, recordemos que dependiendo de lo que se desea hacer se podría utilizar una base polinomial o normal óptima o normal generada por los períodos de Gauss (vistas en los capítulos 2 y 3).

La implementación eficiente de este cifrado, depende de la rapidez del cálculo de la aritmética de puntos racionales de una curva elíptica y a su vez esta del tiempo en que se calculan sumas, multiplicaciones y exponenciaciones sobre el campo finito en la que esta definida la curva.

Más información acerca de criptosistemas sobre curvas elípticas se puede encontrar en [Me93, MeVa93]

Conclusiones

- La suma de elementos sobre el campo finito $\mathbb{F}_{q^n}$ tiene la misma complejidad, esta es $O(n)$ no importando cual sea la base elegida.

- Notamos que tanto para las *bases polinomiales* como para las *bases normales*, realizar la multiplicación depende del desarrollo de algoritmos de multiplicación rápida de polinomios. Un factor importante para que la multiplicación de polinomios, es que estos sean sobre $\mathbb{F}_q[\beta]$ donde β es una raíz n-ésima primitiva de la unidad en alguna extensión de $\mathbb{F}_q$ ya que estas proporcionan muchas ventajas, tales como la disminución de operaciones cuando se hace la sustitución $\beta^n = 1$.

- En el caso de los campos finitos $\mathbb{F}_{2^n}$ son mas usadas en la práctica las bases polinomiales, como en el caso del sistema de cifrado Rijndael.

- Las complejidades de realizar operaciones son similares, pero la ventaja que ofrecen los períodos de Gauss, es la facilidad para calcular su tabla de multiplicar, recordemos que son cardinalidades de intersecciones de conjuntos.

- Si queremos almacenar la tabla de multiplicar de una *base polinomial* de un campo finito $\mathbb{F}_{q^n}$, tenemos que ocupar lugar para a lo más n^2 valores distintos de cero.

- Cuando usamos una *base normal óptima* sobre un campo finito $\mathbb{F}_{p^n}$ necesitamos almacenar a lo más $2n - 1$ valores distintos de cero, para efectuar la multiplicación entre dos elementos del campo.

- La tabla de multiplicar de los elementos de una *base normal generada por los períodos de Gauss* sobre $\mathbb{F}_{q^n}$ tiene a lo más nk valores t_{ij} distintos de cero.

- El método para construir *bases normales* por medio los períodos de Gauss del tipo (n, k) es más eficiente cuando k es pequeño.

- Las condiciones para construir períodos de Gauss se pueden generalizar para un entero arbitrario r tal que $\phi(r) = nk$. Esta situación no se describe en este trabajo pero se puede consultar en el artículo *Normal Bases via Gauss periods* [FeGaSh98].

Apéndice A

Complejidad Computacional

Un *algoritmo* es un conjunto de instrucciones sencillas, claramente especificado, que se debe seguir para resolver un problema [AhHoUl74, We94]. Una vez que se da un algoritmo correcto para un problema, es importante determinar la cantidad de recursos, digamos tiempo o espacio, que usará. Veremos como calcular el tiempo requerido.

Definición A.0.4. $T(n) = O(f(n))$ *para la función* $f : \mathbb{N} \mapsto \mathbb{R}$, *si existen constantes* c *y* n_0 *tales que* $T(n) \leq c\, f(n)$ *cuando* $n \geq n_0$.

Definición A.0.5. $T(n) = \Omega(g(n))$ *para la función* $g : \mathbb{N} \mapsto \mathbb{R}$, *si existen constantes* c *y* n_0 *tales que* $T(n) \geq c\, g(n)$ *cuando* $n \geq n_0$.

Definición A.0.6. $T(n) = \Theta(h(n))$ *si y sólo si* $T(n) = O(h(n))$ *y* $T(n) = \Omega(h(n))$ *para la función* $h : \mathbb{N} \mapsto \mathbb{R}$,.

Definición A.0.7. $T(n) = o(h(n))$ *si* $T(n) = O(p(n))$ *y* $T(n) \neq \Omega(g(n))$.

Notemos que aunque $1000n$ es mayor o igual que n^2 para $n \leq 1000$, n^2 crece más rápido que $1000n$, así $n^2 > 1000n$ cuando $n \mapsto \infty$. Siendo el punto de cambio $n = 1000$. La definición A.0.4 dice que existe un punto n_0 tal que $c \cdot f(n)$ es al menos tan grande como $T(n)$, de tal modo que se ignoran los valores constantes, entonces $f(n)$ es al menos tan grande como $T(n)$. En este caso tenemos $T(n) = 1000n$, $f(n) = n^2$, $n_0 = 1000$ y $c = 1$. Se podría usar $n_0 = 10$ y $c = 100$. Así podemos decir que $1000n = O(n^2)$ llamado orden n cuadrada. Esta notación se conoce como O grande.

Para demostrar que alguna función $T(n) = O(f(n))$, no se aplican las definiciones formalmente, sino que se usan resultados conocidos. Cuando decimos que $T(n) = O(f(n))$, decimos que la función $T(n)$ crece a una velocidad no mayor que $f(n)$; así $f(n)$ es una *cota superior* de $T(n)$. Como esto implica $f(n) \geq T(n)$ entonces se cumple que $f(n) = \Omega(T(n))$, por lo que $T(n)$ es una *cota inferior* de $f(n)$.

Por ejemplo, n^3 crece más rápido que n^2, así podemos decir que $n^2 = O(n^3)$ o que $n^3 = \Omega(n^2)$. $f(n) = n^2$ y $g(n) = 2n^2$ crecen a la misma velocidad, así ambas satisfacen que $f(n) = O(g(n))$ y $f(n) = \Omega(g(n))$. Cuando escribimos $g(n) = \Theta(n^2)$, no sólo afirmamos que $g(n) = O(n^2)$, sino también que el resultado es tan exacto como es posible.

Usaremos los siguientes lemas para determinar y ordenar las tasas de crecimiento.

Lema A.0.8. *Si* $T_1(n) = O(f(n))$ *y* $T_2(n) = O(g(n))$, *entonces*

(a) $T_1(n) + T_2(n) = \text{máx}(O(f(n)), O(g(n)))$,

(b) $T_1(n) \cdot T_2(n) = O(f(n) \cdot g(n))$.

Demostración. Por definicón existen cuatro constances c_1, c_2, n_1 y n_2, tales que $T_1(n) \leq c_1 f(n)$ para $n \geq n_1$ y $T_2(n) \leq c_2 g(n)$ para $n \geq n_2$.
(*a*) Sea $n_0 = \text{máx}(n_1, n_2)$, entonces para $n \geq n_0$, también se cumple $T_1(n) \leq c_1 f(n)$ y $T_2(n) \leq c_2 g(n)$, por lo que $T_1(n) + T_2(n) \leq c_1 f(n) + c_2 g(n)$. Sea $c_3 = \text{máx}(n_1, n_2)$, entonces,

$$\begin{aligned} T_1(n) + T_2(n) &\leq c_3 f(n) + c_3 g(n) \\ &\leq c_3 (f(n) + g(n)) \\ &\leq 2c_3 \,\text{máx}(f(n), g(n)) \\ &\leq c \,\text{máx}(f(n), g(n)) \end{aligned}$$

$$para \quad c = 2c_3 \; y \; n \geq n_0.$$

(*b*) Para este caso se cumple que $T_1(n)T_2(n) \leq c_1 f(n) c_2 g(n) = c_1 c_2 f(n) g(n)$, entonces sólo debemos considerar $c = c_1 c_2$. □

Lema A.0.9. *Si* $T(x)$ *es un polinomio de grado* n, *entonces* $T(x) = \Theta(x^n)$.

Función	Nombre
c	constante
$log\, n$	logarítmica
$log^2 n$	logarítmica cuadrada
n	lineal
n^2	cuadrática
n^3	cúbica
2^n	exponencial

Tabla A.1: Tasas de crecimiento características.

Demostración. Primero notemos que $x = O(x)$. Supongamos cierto que $x^k = O(x^k)$. Ahora $x^{k+1} = x^k x = O(x^{k+1}x) = O(x^{k+1})$ por el inciso (b) del lema A.0.8, se cumple entonces para cualquier n.

Si consideramos el polinomio $T(x) = \sum_{i=0}^{n} a_i x^i$

$$T(x) = \text{máx}(O(a_n x^n), O(a_{n-1}x^{n-1}, \ldots, O(a_0))) = O(x^n)$$

por el inciso (a) del lema A.0.8, así $T(x) = O(x^n)$, análogamente podemos obtener $T(x) = \Omega(x^n)$, por lo que $T(x) = \Theta(x^n)$. □

Lema A.0.10. *$log^k n = O(n)$ para cualquier k constante. Esto indica que los logaritmos crecen muy lentamente.*

En la tabla A.1 vemos las tasas de crecimiento más comunes, y en la figura A.1, podemos compararlas.

Debemos notar que, no se debe incluir constantes o términos de orden menor en una O grande. No se debe escribir $T(n) = O(5n^2)$ o $T(n) = O(n^2 + n)$, en ambos casos la forma correcta es $T(n) = O(n^2)$. Es decir, cuando calculamos O grande debemos simplificar.

Para analizar algoritmos, se considera que las operaciones tales como suma, multiplicación o lectura a disco toman el mismo tiempo, aunque esto en la realidad no es verdad. También se considera la entrada, buscando una función que defina el tiempo de ejecución quedandonos con el peor caso.

Ejemplo A.0.11. *Consideremos simplemente la suma de cuadrados de n elementos. La entrada es n y por lo tanto se realizan n sumas y n multi-*

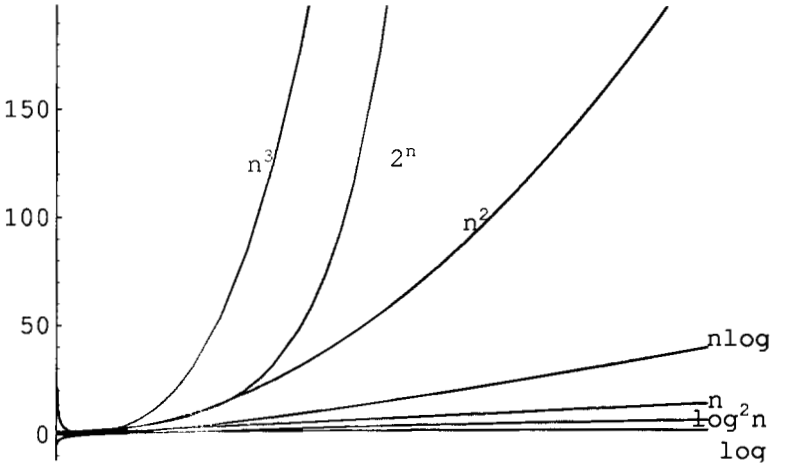

Figura A.1: Tasas de crecimiento

plicaciones, con lo que resultan ser en total $2n$ *operaciones, entonces, la* O *grande es* $O(n)$.

Bibliografía

[Ag88] Agnew G., Mullin R., Vanstone S., *Fast exponentiantion in* $\mathbb{F}_{2^n}$. EUROCRYPT 88, LNCS 330 (1988), 251-255.

[AhHoUl74] Aho A. V., Hopcroft J.E., Ullman J.D., *The Design and Analysis of Computer Algorithms.* Addison-Wesley, (1974).

[AsBlVa89] Ash D. W., Blake I. F., Vanstone S. A., *Low Complexity Normal Bases.* Discrete Applied Mathematics 25 (1989), 191-210.

[BlMi84] Blum M., Micali S., *How to generate cryptographically strong sequences of pseudorandom bits.* SIAM J. Computing 13 (1984), 850-864.

[BoKu95] Bocharova I. E., Kudrynahov B. D., *Fast Exponentation in Cryptography.* LNCS 948 (1995),146-157.

[BrGoMcWi92] Brickell E. F., Gordon D. M., McCurley K. S., Wilson D. B., *Fast Exponentiation with Precomputation.* EUROCRYPT 92 LNCS (1992), 200-207.

[Ca89] Cantor D. G., *On Arithmetical Algorithms over Finite Fields.* Journal of Combinatorial Theory, Series A 50 (1989), 285-300.

[CaKa91] Cantor D. G., Kaltofen E., *On fast multiplication of polynomials over arbitrary algebras.* Acta Informatica 28 (1991), 693-701.

[Ca91] Cassels J.W.S., *Lectures on Elliptic Curves.* Cambridge University Press (1991).

[DaRi98] Daemen J., Rijmen V., *AES Proposal: Rijndael.* NIST AES Proposal, Jun (1998), http://www.esat.kuleuven.ac.be/ rijmen/rijndael/index.html.

[DiHe76] Diffie W., Hellman M.E., *New Directions in Cryptography.* IEEE Transactions on Information Theory, Vol. IT-22 (1976), 644-654.

[El85] ElGamal T., *A public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms.* IEEE Transactions on Information Theory, Vol. IT-31 (1985), 469-472.

[FeGaSh98] Feistel S., von zur Gathen J., Shokrollahi M. A., *Normal bases via General Gauss Periods.* http://math-www.uni-paderborn.de/ aggathen/index.html

[FIPS46] *Data enctyption standard.* Federal Information Processing Standard Publication 46. NIST 1977.

[GaGaPa95] Gao S., von zur Gathen J., Panario D., *Gauss Periods and fast exponentation in finite fields.* LATIN 95, LNCS 911 (1995), 343-352.

[GaGaPa98] Gao S., von zur Gathen J., Panario D., *Gauss Periods, Orders and Cryptographical Applications.* Mathematics of Computation Vol.67, Num.221 (1998), 343-352.

[GaGaPaSh] Gao S., von zur Gathen J., Panario D., Shoup V.. *Algorithms for exponentiation in finite fields.* http://shoup.net/papers

[GaVa95] Gao S., Vanstone S. A., *On Orders of Optimal Normal Basis Generators.* Mathematics of Computation Vol. 64, No. 211, (1995), 1227-1233.

[GaNo97] von zur Gathen J., Nöcker M., *Exponentiation in Finite Fields: Theory and Practice.* AAECC-12, LNCS 1255 (1997), 88-133.

[GaSh95] von zur Gathen J., Shparlinski I., *Orders of Gauss Periods in Finite Fields.* ISAAC 95, LNCS 1004 (1995), 208-215.

[Ga86] Gauss C.F., *Disquisitiones Arithmeticae.* Braunscheweig 1801. Springer Verlag (1986).

[GeCzLa] Geddes K. O., Czapor S. R., Labahn G., ***Algorithms for Computer Algebra.*** Kluwer Academic Publishers.

[Ha67] Hall, M. Jr., *Combinatorial Theory.* Blaisdell, (1967).

[ItTs88] Itoh T., Tsujii S., *A Fast Algorithm for Computing Multiplicative Inverses in $GF(2^m)$ Using Normal Bases.* Information and Computation Vol. 78, No. 3 (1988), 171-177.

[Ka67] Kahn D., *The Codebreakers.* Macmillan (1967).

[Kn81] Knuth D. E., *The Art of Computer Programming* Vol. 2, Addison-Wesley (1981).

[Ko87] Koblitz N., *Elliptic Curve Cryptosystems.* Mathematics of Computation Vol. 48 (1987), 203-209.

[Ko87a] Koblitz N., *A Course in Number Theory and Cryptography.* Springer Verlag, New York, (1987).

[LeSc87] Lenstra H.W., Schoof R.J., *Primitive Normal Bases for Finite Fields.* Mathematics of Computation Vol.48, No.177 (1987), 217-231.

[LiNi83] Lidl R., Niederreiter H., *Finite Fields.* Addison-Wesley Publishing Company (1983).

[MaSl77] MacWilliams F. J., Sloane N. J. A., *The Theory of Error-correcting Code.* NorthHolland, (1977).

[Mc82] McEliece R. J., *Finite Fields for Computer Scientist and Engineers.* Kluwer Academic Publishers (1982).

[Me93] Menezes A. J., *Elliptic Curve Public Key Cryptosystems.* Kluwer Academic Publishers (1993).

[MeBlGa93] Menezes A. J., Blake I., Gao X., et. al., *Applications of Finite Fields.* Kluwer Academic Publishers (1993).

[MeOoVa96] Menezes A. J., van Oorschot P. C., Vanstone S. A., *Handbook of Applied Cryptography.* CRC press (1996).

[MeVa93] Menezes A. J., Vanstone S. A., *Elliptic curve cryptosystems and their implementation.* Journal of Cryptology 6 (1993), 209-224.

[Mi86] Miller V. S., *Use of Elliptic Curves in Cryptography.* Advances in Cryptology-CRYPTO'85 LNCS (1986), 417-426.

[MuOnVa89] Mullin R. C., Onyszchuk J. M., Vanstone S. A., Wilson R. M., *Optimal Normal Bases in $GF(p^n)$.* Discrete Applied Mathematics 22 (1989), 149-161.

[MuSh] Mullen G. L., Shparlinski I., *Open problems and Conjectures in Finite Fields.*

[NiZu85] Niven I., Zuckerman H. S., *Introducción a la Teoría de Números.* Limusa, México D.F. (1985).

[Ny94] Nyberg K., *Differential uniform mappings for cryptography.* LNCS 765 (1994), 55-64.

[PaSo97] Paar C., Soria-Rodriguez P., *Fast Arithmetic Architectures for Public-Key Algorithms over Galois Fields $GF((2^n)^m)$.* LNCS 1233 (1997), 363-378.

[Pl98] Pless V., *Handbook on Coding.* Elsevier Publishing Co. (1998).

[RiShAd78] Rivest R., Shamir A., Adleman L., *A Method for Obtaining Digital Signatures and PKC.* Communications of the ACM, Vol. 21, No. 2 (1978), 120-128.

[Ro94] de Rooij P., *Efficient Exponentiation using Precomputation and Vector Addition Chains.* LNSC 950 (1994), 389-399.

[Ry63] Ryser H. J., *Combinatorial Mathematics.* Carus. Math. Monographs, No. 14, Math. Assoc. of America, New York (1963).

[Sc75] Schönhage A., *A Lower Bound for the Length of Addition Chains.* Theorical Computer Science 1 (1975), 1-12.

[Sc77] Schönhage A., *Schenelle Multiplikation von Polynomen über Körpen der Charakteristik 2.* Acta Informatica 7 (1977), 395-398.

[Sh] Shparlinski I., *Approximate constructions in Finite Fields.* http://www.comp.mq.edu.au/ igor/

[Sh92] Shparlinski I., *Computational and Algorithmic Problems in Finite Fields.* Kluwer Academic Publishers (1992).

[Si86] Silverman J., *The Arithmetic of Elliptic Curves.* Springer Verlag, New York (1986).

[St95] Stinson D., *Cryptography: Theory and Practice.* CRC Press (1995).

[St90] Stinson D. R., *Some Observations on Parallel Algorithms for Fast Exponentiation in* $GF(2^n)^*$. SIAM Journal of Computation Vol.19, No.4 (1990), 711-717.

[WaStWa72] Wallis W. D., Street A. P., Wallis J. S., *Combinatorics: Room Squares, Sum-Free Sets, Hadamard Matrices.* Lecture Notes in Math., Vol. 292 (1972).

[Wa89] Wang C., *An Algorithm to Design Finite Field Multipliers Using a Self-Dual Normal Basis.* IEEE Transactions on Computers Vol. 38, No.10 (1989), 1457-1460.

[WaBl97] Wang M., Blake I. F., *Normal Basis of the Finite Field* $\mathbb{F}_{2(p-1)p^m}$ *over* $\mathbb{F}_2$. IEEE Transactions on Information Theory, Vol. 43, No. 2 (1997), 737-739.

[Wa82] Washington L. C., *Introduction to Cyclotomic Fields.* Springer Verlag, New York, (1982).

[Wa90] Wassermann A., *Konstruktion von Normalbasen.* Bayreuther Mathematische Schriften 31 (1990), 155-164.

[We94] Weiss M. A., *Estructuras de Datos y Algoritmos.* Addison-Wesley, (1994).

[WiBoVaGe96] De Win E., Bosselaers A., Vandenberghe S., De Gersem P., Vanderwalle J., *A Fast Software Implementation for Arithmetic Operations in* $GF(2^n)$. LNCS 1233 (1996), 65-76.

[Ya90] Yacobi Y., *Exponentiating Faster with Addition Chains.* LNCS 473 (1990), 222-229.

Índice de Materias