UNIVERSITAT JAUME I

Máster en matemática computacional

Trabajo de fin de máster

Códigos lineales correctores de evaluación: criptosistema de McEliece

Alumno
José Ever
Gonzales Euceda

Tutor
Dr. Carlos
Galindo Pastor

Curso académico: 2017-2018

A mi familia

A mi esposa y mis hijos.

Aún hay sangre en mis venas.

Agradecimientos

Doy gracias a Dios por permitirme iniciar, vivir y terminar esta experiencia educativa.

Agradezco a toda mi familia por cada sacrificio que hacen por mí.

Agradezco a la universidad de Jaume I por permitirme ser parte de ella y a todos los profesores del máster que me brindaron sus conocimientos y a las personas que de alguna forma colaboraron en mi formación.

Agradezco al doctor Carlos Galindo Pastor por su dedicación, por su tiempo hasta de madrugada, por su gran disposición de ayuda inclusive hasta cuando estaba de vacaciones, por su paciencia y por guiarme en el gran mundo de la criptología.

Agradezco al doctor Vicente Martínez por ayudarme durante todo el máster y por cada que vez que me dijo ¡ánimo!.

Índice general

Capítulo 1

Introducción

Es gloria de Dios tener secretos, y honra de los reyes penetrar en ellos.
Proverbios *25.2*

Desde un principio, el ser humano ha sentido la necesidad de tener secretos. Tan solo en algunas situaciones deseaba compartirlos con sus amigos o aliados. Esta necesidad a lo largo de la historia le ha servido al hombre para potenciar su ingenio con el fin de proteger sus secretos. Ha desarrollado métodos que le permiten ocultarlos de los que consideraba enemigos, pero que le permiten a sus amigos o aliados tener rápido acceso a ellos. Sin embargo, también se ha potenciado el ingenio de muchos humanos con el fin de desarrollar métodos que permita tener acceso a información privada o confidencial.

En consecuencia, los seres humanos sienten la necesidad de ocultar lo mejor posible toda aquella información que sea considerada privada, a fin de mantenerla a salvo de intrusos que pueden incluso llegar a hacer un mal uso de ella. Y así nace la Criptología.

El siguiente trabajo presenta algunos aspectos actuales de este problema. Ya no están los antiguos actores como los espartanos con *La scitala*, Julio Cesar con su criptosistema de desplazamiento, los antiguos cristianos atribuyendo el número 666 al emperador Nerón, las mujeres hindú aprendiendo criptografía del Kama Sutra, el mismo Isaac Newton que al parecer estaba convencido de que la Biblia ocultaba un código capaz de revelar el futuro o Alan Turing queriendo descifrar el código Enigma. Actualmente ha entrado en escena un contendiente que promete revelar los secretos de todos, *el computador cuántico*.

El primer capítulo trata acerca de los sistemas criptográficos de clave pública más utilizados actualmente, estos son el criptosistema RSA y el criptosistema de ElGammal con sus variantes. Presentamos sus características y los ataques mas conocidos a estos.

El segundo capítulo trata acerca de los códigos lineales de evaluación correctores y conceptos fundamentales para estos códigos como lo es las funciones de orden, funciones grado y funciones peso. Además, de las propiedades de estos códigos y de algoritmos de decodificación. Estos algoritmos resultan ser eficientes. Y como se verá pueden ser una posible solución temporal a este problema y equilibrar la contienda.

El tercer capítulo introduce el criptosistema de McEliece, el cual resulta ser una posible esperanza para que los secretos sigan siendo secretos, ya que posee un sistema seguro contra ataques cuánticos. Esta seguridad descansa en la familia de códigos lineales a utilizar y he aquí donde los códigos lineales de evaluación correctores surgen como una posible buena opción.

Capítulo 2

Criptosistemas de clave pública

2.1. Introducción

En este capítulo se hace una breve introducción a los sistemas criptográficos de clave pública más utilizados en la actualidad [Stinson, 2005], estos son el sistema RSA y ElGamal, ambos basan su seguridad en problemas de la teoría de números cuyas soluciones son computacionalmente inviables.

Damos a continuación varias definiciones que permiten introducir estos criptosistemas.

Definición 2.1 Una función en una vía es una aplicación biyectiva $f : A \to B$ tal que $\forall x \in A$ el cálculo de $f(x)$ se puede efectuar en tiempo polinómico, pero dado $y \in B$ es, por lo general, computacionalmente infactible determinar $x \in A \ / \ f(x) = y$, sin información adicional.

Definición 2.2 Una función trampa es una función en una vía, con un elemento adicional t (información secreta o trampa) tal que con este t es posible generar un algoritmo que en tiempo polinómico es factible determinar el $x \in A \ / \ f(x) = y$.

Definición 2.3 Un criptosistema es una 5-upla $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ donde las siguientes condiciones son satisfechas:

1. $\mathcal{P}$ es un conjunto finito de texto a encriptar.

2. $\mathcal{C}$ es un conjunto finito de texto encriptado.

3. $\mathcal{K}$, espacio de claves, es un conjunto finito de posibles claves a utilizar.

4. $\forall K \in \mathcal{K}$ existe una regla de encriptación $e_k \in \mathcal{E}$ y una correspondiente regla de desencriptación $d_k \in \mathcal{D}$. Cada $e_k : \mathcal{P} \to \mathcal{C}$ y $d_k : \mathcal{C} \to \mathcal{P}$ son funciones tales que $d_k(e_k(x)) = x \ \forall x \in \mathcal{P}$.

Definición 2.4 Un criptosistema se denomina computacionalmente seguro cuando el cálculo de la regla de desencriptación d_k no es factible.

Definición 2.5 Un criptosistema de clave pública es aquel criptosistema donde e_k es una función trampa f y $d_k(y) = e_k(y, t)$ tal que, $y \in \mathcal{C}$ y t es la información privada.

Dentro de los sistemas de clave pública, entre los más importantes [Stinson, 2005] tenemos:

1. RSA: Su seguridad está basada en la dificultad de factorizar números muy grandes.

2. ElGamal: Su seguridad está basada en el problema del logaritmo discreto.

Revisamos brevemente a continuación el criptosistema RSA.

2.2. Criptosistema RSA

El siguiente cuadro muestra la 5-tupla que define el sistema RSA.

Criptosistema 2.6 Sea n un entero positivo que se define como $n = pq$, donde p y q son números primos. Sea $\mathcal{P} = \mathcal{C} = \mathbb{Z}_n$.El criptosistema RSA se define

$$\mathcal{K} = \{(n, p, q, a, b) : ab \equiv 1 \pmod{\phi(n)}\},$$

donde ϕ es la función ϕ de Euler, y las funciones de cifrado y descifrado son

$$e_k(x) = x^b \pmod{n}$$

y

$$d_k(y) = y^a \pmod{n}$$

$(x, y \in \mathbb{Z}_n)$. Los valores (n, b) comprenden la clave pública, y los valores (p, q, a) forman la clave privada.

2.2.1. Exactitud del criptosistema RSA

El siguiente resultado garantiza que el criptosistema RSA es correcto.

Teorema 2.7 Sean las funciones de encriptación e y desencriptación d definidas como en el **Criptosistema** 2.6, al igual que el número n y $x \in \mathcal{P}, y \in \mathcal{C}$, entonces:

$$(x^e)^d = x.$$

Prueba: Si $\mathbb{Z}_n^*$ es el conjunto de residuos módulo n que son relativamente primos a n y siendo que $ab \equiv 1 \pmod{\phi(n)}$, se tiene que $ab = t\phi(n) + 1$ para algún entero $t \geq 1$. Si $x \in \mathbb{Z}_n^*$, entonces

$$\begin{aligned}(x^b)^a &\equiv x^{t\phi(n)+1} \pmod{n} \\ &\equiv \left(x^{\phi(n)}\right)^t x \pmod{n} \\ &\equiv (1)^t x \pmod{n} \\ &\equiv x \pmod{n},\end{aligned}$$

donde la segunda congruencia se obtiene por el Teorema de Euler.

Si $x \in \mathbb{Z}_n \backslash \mathbb{Z}_n^*$, notemos que $\phi(n)$ divide al número $ab - 1$, por lo que $ab - 1 = t\phi(n) = t(p-1)(q-1)$ para algún entero $t \geq 1$, entonces

$$\begin{aligned}(x^b)^a &= x^{ab-1}x \\ &= \left(x^{t\phi(n)}\right)x \\ &= \left(x^{t(p-1)(q-1)}\right)x \\ &= \left(x^{(p-1)}\right)^{t(q-1)}x \\ &= (1)^{(q-1)t}x \pmod{p} \\ &\equiv x \pmod{p},\end{aligned}$$

donde la penúltima igualdad se debe al pequeño Teorema de Fermat. Finalmente si efectuamos el mismo proceso, podemos obtener que $\left(x^b\right)^a = x \pmod{q}$ y por el teorema del residuo chino se obtiene el resultado. □

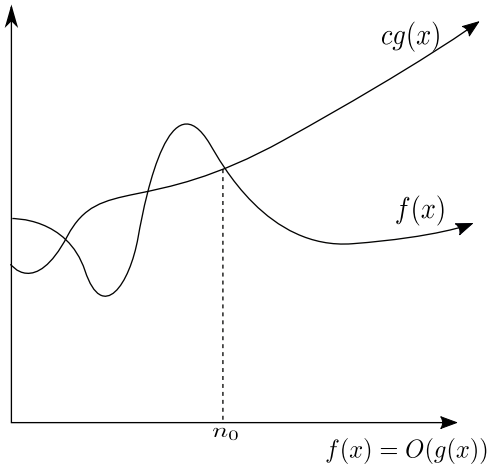
A continuación, damos una rápida introducción a la complejidad de los problemas a resolver mediante algoritmos y su relación con el RSA y logaritmo discreto.

Definición 2.8 El tamaño de entrada de un algoritmo es el número total de bits necesarios para representar la entrada de un algoritmo que manipula enteros en notación binaria.

Definición 2.9 El tiempo de ejecución de un algoritmo para un tamaño de entrada es la cantidad de operaciones primitivas $(+, -, *, /,$ mód $)$ o pasos a ejecutar [Cormen et al., 2009].

La notación asintótica O sirve para describir el tiempo de ejecución asintótico de un algoritmo en términos de funciones cuyo dominio es $\mathbb{N}$ [Cormen et al., 2009].

Definición 2.10 Para una función $g : \mathbb{Z}^+ \to \mathbb{Z}^+$, denotamos por $O(g(n))$ al conjunto de funciones: $O(g(n)) = \{f : \mathbb{Z}^+ \to \mathbb{Z}^+$ tales que existen constantes positivas c y n_0 satisfaciendo $0 \leq f(n) \leq cg(n)$, para todo $n \geq n_0\}$.



Sean x e y enteros positivos teniendo k y l bits respectivamente, entonces $k = \lfloor \log_2 x \rfloor + 1$ y $l = \lfloor \log_2 y \rfloor + 1$.

Asumiendo que $k \geq l$ se tiene que el tiempo de ejecución de las operaciones básicas se pueden desarrollar en notación asintótica O (*big O*) como sigue en la siguiente tabla:

Operación	Tiempo de ejecución
x+y	O(k)
x-y	O(k)
xy	O(kl)
$\left\lfloor \frac{x}{y} \right\rfloor$	O(l(k-l))
$\gcd(x, y)$	O(k^3)

Los problemas computacionales se pueden clasificar en los siguientes cuatro tipos:

1. Problemas solucionables.
2. Problemas irresolubles.
3. Problemas solucionables pero impracticables.
4. Problemas solucionables pero intratables.

Un problema se considera computacionalmente solucionable si existe al menos un algoritmo para resolverlo, un problema para el que no existe un algoritmo conocido (en ese momento) para resolverlo es un problema irresoluble. Algunos problemas solucionables se denominan impracticables debido a la enorme cantidad de recursos computacionales (incluyendo el tiempo de ejecución) necesarios para resolverlos. Estos problemas computacionales son muy simples de caracterizar pero difíciles de resolver y se utilizan técnicas como la heurística para resolverlos. Los problemas que son solucionables y difíciles de resolver se denominan intratables ya que ni con algoritmos más rápidos ni la invención de computadoras más rápidas podrían hacerlos solucionables.

La teoría de la complejidad computacional clasifica estos problemas computacionales de la siguiente manera:

1. **P:** Estos son problemas que pueden resolverse en tiempo polinomial.
2. **NP:** Esto significa "tiempo polinomial no determinista". Un problema está en NP si podemos rápidamente, en tiempo polinomial; probar si una solución es correcta sin preocuparse por lo difícil que puede ser encontrar la solución.
3. **PSPACE:** Problemas que se pueden resolver usando una cantidad razonable de memoria sin importar cuánto tiempo toma la solución.
4. **EXPTIME:** Estos son problemas que pueden resolverse en tiempo exponencial. Esta clase contiene todo en las clases P, NP y PSPACE.
5. **NP-HARD:** Es una clase de problemas que son al menos tan difíciles como los problemas más difíciles en NP.
6. **NP-Completo:** Son los problemas más difíciles en NP. Si alguien encuentra un algoritmo de tiempo polinomial para incluso un problema P-Completo, eso implicaría un algoritmo de tiempo polinomial para cada problema NP-completo.
7. **BQP:** Son los problemas que las computadoras cuánticas pueden resolver de manera eficiente y significa "error acotado, cuántico, tiempo polinómico"(por sus siglas en inglés).

Tanto la factorización de enteros (en la que se basa el criptosistema RSA) como el cálculo del logaritmo discreto tienen complejidad BQP. Ambos problemas son NP-HARD [Okeyinka, 2017].

Definición 2.11 Un algoritmo aleatorio es cualquier algoritmo que utiliza números aleatorios y un algoritmo determinista es un algoritmo que no es aleatorio.

Definición 2.12 Un problema de decisión es un problema en el cual una pregunta se responde con un **Si** o un **No**.

En la búsqueda de números primos grandes lo que evita la factorización algorítmica de n, el problema de desición que se aplica de hecho es el siguiente.

Nombre:	Compuesto
instancia	$n \in \mathbb{N}, n \geq 2$
Pregunta:	¿Es n un número compuesto?

Mostramos a continuación la implementación del criptosistema RSA.

2.2.2. Implementación del criptosistema RSA

Algoritmo 2.1: Implementación de RSA

1 Generar dos primos grandes, p y q, tales que $p \neq q$;
2 $n \leftarrow pq$ y $\phi(n) \leftarrow (p-1)(q-1)$;
3 Seleccionar aleatoriamente $b(1 < b < \phi(n))$ tal que $\gcd(b, \phi(n)) = 1$;
4 $a \leftarrow b^{-1} \pmod{\phi(n)}$;
5 La clave pública es (n, b) y la clave privada es (p, q, a).

Definición 2.13 El algoritmo de Montecarlo es un algoritmo aleatorio para un problema de decisión para el que una respuesta SI es correcta, pero una respuesta NO puede ser incorrecta, por lo que decimos que este algoritmo tiene error de probabilidad ϵ si para cualquier instancia en la que la respuesta es Si el algoritmo podría dar la respuesta incorrecta NO con probabilidad a los más ϵ.

Como hemos dicho, parte del algoritmo de la implementación del criptosistema RSA se basa en la generación de números primos aleatorios grandes. El método a utilizar es generar números aleatorios grandes y aplicarles un **test de primalidad** a través de la utilización de un algoritmo aleatorio de Montecarlo de tiempo polinómico. Estos algoritmos permiten decidir, como hemos mencionado, si un número entero es compuesto. Ejemplos de ellos son el **algoritmo SOLOVAY-STRASSEN** y **el algoritmo de MILLER-RABIN** (conviene aclarar que existe un algoritmo determinista).

A continuación ofrecemos información para poder establecer el primero de los algoritmos anteriores.

Definición 2.14 Sea p un primo impar y a un entero. Se dice que a es un residuo cuadrático módulo p si $a \not\equiv 0 \pmod{p}$ y la congruencia $y^2 \equiv a \pmod{p}$ tiene una solución $y \in \mathbb{Z}_p$.

Teorema 2.15 Sea p un número primo impar. Entonces a es un residuo cuadrático módulo p si y solo si:

$$a^{(p-a)/2} \equiv 1 \pmod{p}.$$

Una prueba de este teorema se puede encontrar en la página 180 de [Stinson, 2005].

Definición 2.16 Sea p un número primo impar. Para cada entero a, definimos los símbolos de Legendre $\left(\frac{a}{p}\right)$ como sigue:

$$\left(\frac{a}{p}\right) = \begin{cases} 0 & \text{si } a \equiv 0 \pmod{p}, \\ 1 & \text{si } a \text{ es un residuo cuadrático módulo } p, \\ -1 & \text{si } a \text{ no es un residuo cuadrático módulo } p. \end{cases}$$

Definición 2.17 Sea n un número primo impar, y

$$n = \prod_{i=1}^{k} p_i^{e_i}$$

su factorización en potencias de primos.

Si a es un entero, el símbolo de Jacobi $\left(\frac{a}{n}\right)$ está definido como:

$$\left(\frac{a}{n}\right) = \prod_{i=1}^{k} \left(\frac{a}{p_i}\right)^{e_i}.$$

Listamos a continuación algunas propiedades del símbolo de Jacobi.

1. Si n es un número primo impar, entonces el símbolo de Jacobi es el correspondiente símbolo de Legendre.
2. Si $a \equiv b \pmod{n}$, entonces $\left(\frac{a}{n}\right) = \left(\frac{b}{n}\right)$.
3. $\left(\frac{a}{n}\right) = \begin{cases} 0 & \text{si } \gcd(a,n) \neq 1, \\ \pm 1 & \text{si } \gcd(a,n) = 1. \end{cases}$
4. $\left(\frac{ab}{n}\right) = \left(\frac{a}{n}\right)\left(\frac{b}{n}\right)$ si n es un entero positivo impar.

5. $\left(\frac{2}{n}\right) = \begin{cases} 1 & \text{si } n \equiv \pm 1 \pmod 8, \\ \pm 1 & \text{si } n \equiv \pm 3 \pmod 8). \end{cases}$

6. Si m y n son primos impares, entonces $\left(\frac{m}{n}\right) = \begin{cases} -\left(\frac{n}{m}\right) & \text{si } m - n \equiv \pmod 4, \\ \left(\frac{n}{m}\right) & \text{en otro caso.} \end{cases}$

Estas propiedades permiten generar un algoritmo que calcule los símbolos de Legendre en tiempo polinomial (Ver capítulo 5).

Mostramos a continuación el algoritmo de Solovay-Strassen. Este es un algoritmo de Montecarlo con una probabilidad de error a lo más de 1/2 [Stinson, 2005].

Algoritmo 2.2: SOLOVAY-STRASSEN

Input: $n \in \mathbb{Z}^+$.
1 Seleccionar un número entero a tal que $1 \leq a \leq n-1$.
2 $x \leftarrow \left(\frac{a}{n}\right)$.
3 **if** $x = 0$ **then**
4 **return** ("n ES COMPUESTO").
5 $y \leftarrow a^{(n-1)/2} \pmod n$.
6 **if** $x \equiv y \pmod n$ **then**
7 **return** ("n ES PRIMO").
8 **else**
9 **return** ("n ES COMPUESTO").

El algoritmo de Solovay Strassen en su segunda línea de instrucciones calcula el símbolo de Jacobi. Para calcularlo se puede utilizar el algoritmo propuesto en el capítulo 5 que se ejecuta en tiempo polinomial. Se puede utilizar el algoritmo **SQUARE-AND-MULTIPLE** (ver capítulo 5) para la exponenciación modular y **EL ALGORITMO DEL MULTIPLICATIVO INVERSO** para calcular $\left(\frac{a}{b}\right) = a^{(n-1)/2} \pmod n$ (instrucción 6 del algoritmo puesta como resultado) ya que la congruencia $ax \equiv b \pmod m$ tiene una única solución $x \in \mathbb{Z}_n$ si y solo si $\gcd(a, m) = 1$ [Stinson, 2005]. Finalmente para determinar el máximo común divisor se puede utilizar el **ALGORITMO EXTENDIDO DE EUCLIDES** (ver capítulo 5).

Indicamos ahora que en el algoritmo extendido de Euclides que mostramos en el capítulo 5, todas las instrucciones exceptuando las instrucciones 7 y 18 se efectúa en tiempo constante $O(1)$. La instrucción 7 y 18 al colocar ambas representaciones binarias de igual tamaño, esto es $l = k$, se efectúan en tiempo $O(k^2) = O(\log(x)^2)$ (ver teoría de tiempo de ejecuciones). El ciclo de repetición WHILE de la instrucción 9 se efectúa en un número constante de veces por lo que el algoritmo se efectúa en tiempo $O(\log(x)^2)$ [Binet, 1841]. Una demostración más detallada se puede ver en [Bach and Shallit, 1996] y en [Pommerening, 2016].

El algoritmo Square and Multiply que mostramos en el capítulo 5 asume que el exponente c está representado en notación binaria, esto es

$$c = \sum_{i=0}^{l-1} c_i 2^i,$$

donde l es el número de bits en la representación binaria de c. El método está basado en el hecho de que para todo entero positivo n, nosotros tenemos que

$$x^n = \begin{cases} x\left(x^2\right)^{\frac{n-1}{2}} & \text{si } n \text{ es impar,} \\ \left(x^2\right)^{\frac{n}{2}} & \text{en otro caso ,} \end{cases}$$

por lo que el algoritmo se efectúa en $O((l-1)\log(n)^2)$ operaciones básicas lo que nos lleva a dar una cota superior $O(\log(n)^3)$.

El algoritmo extendido de Euclides produce el valor de $b^{-1} \pmod a$, si este existe. Sin embargo una forma más eficiente de calcular esto, es utilizar el algoritmo del multiplicativo inverso que mostramos también en el capítulo 5.

Todo lo anterior prueba que la implementación del algoritmo RSA se puede ejecutar en $O((\log(n)^3))$ operaciones básicas ya que está acotado superiormente por el cálculo del inverso multiplicativo módulo n.

Otro algoritmo que es util para la prueba de primalidad es **el algoritmo de Miller-Rabin.** Este es un algoritmo de Montecarlo con una probabilidad de error a lo más del 0.25 [Conrad, 2017]. En este algoritmo se asume que n es impar y $\gcd(n, a) = 1$, con esto el algoritmo de Solovay-Strassen tiene una cota superior de ejecución de $O(\log(n)^3)$, pero en la práctica se utiliza más el algoritmo de Miller-Rabin [Koblitz, 1987].

Algoritmo 2.3: Algoritmo de Miller Rabin

Input: $n \in \mathbb{Z}^+$.
Escribir $2^k m = n - 1$, donde m es impar.
Seleccionar un número entero aleatorio a de $[1, n-1]$.
$b \leftarrow a^m \pmod n$.
if $b \equiv 1 \pmod n$ **then**
  return ("n es primo").
for $i \leftarrow 0$ **to** $k-1$ **do**
  if $b \equiv -1 \pmod n$ **then**
    return ("n es primo").
  else
    $b \leftarrow b^2 \pmod n$
return ("n es compuesto").

2.2.3. Ataques al criptosistema RSA

Una forma de quebrantar la seguridad del criptosistema RSA es factorizar el valor n de la clave pública (n, b) y obtener los valores de los primos p, q. Para que el criptosistema RSA sea seguro deben de seleccionarse p y q tales que n sea lo suficientemente grande para que su factorización no sea fáctible.

Ataque de fuerza bruta

El siguiente resultado es obvio.

Teorema 2.18 Si $n \in \mathbb{Z}^+$ y n es compuesto, entonces existe un primo p tal que $p|n$.

La criba de Eratóstenes (o método de la división de prueba) es el método clásico de buscar números primos y se basa en el siguiente resultado.

Teorema 2.19 Si $n \in \mathbb{Z}^+$ y n es compuesto, entonces existe un primo p tal que $p|n$ y $p \leq \sqrt{n}$.

Prueba: Escribamos $n = pq$ tal que $1 < p < n$ y $1 < q < n$ por lo que p o q deben de ser menor o igual a $\sqrt{n}$, sea $p \leq \sqrt{n}$. Si p es primo, culmina la demostración, ahora si p no es primo, entonces existe un primo $a < n$ tal que $a|p$. Por lo que $a|n$ y $a \leq \sqrt{n}$. □

Definición 2.20 El método de la división de prueba consiste en dividir n por cada valor en el intervalo $[2, \lfloor\sqrt{n}\rfloor]$ para decidir si n es primo.

Otros ataques a RSA

El conocido **algoritmo de Pollard p-1** que data del año 1974 es otro algoritmo que se puede emplear para factorizar al número n. Los números primos p y q en RSA también deben elegirse con las propiedades que $p \pm 1$ y $q \pm 1$ tienen al menos un factor primo mayor que 1020, de lo contrario, p podría encontrarse de manera eficiente mediante el uso del algoritmo de factorización de Pollard $p-1$ y el algoritmo de factorización $p+1$ de Williams [Yan, 2007].

Algoritmo 2.4: Algoritmo de factorización de Pollard p-1

Input: $n \in \mathbb{Z}^+$, B: cota superior de búsqueda.

1 $a \leftarrow 2$.
2 **for** $j \leftarrow 2$ **to** B **do**
3 $\quad a \leftarrow a^j \pmod{n}$.
4 $d \leftarrow \gcd(a-1, n)$.
5 **if** $1 < d < n$ **then**
6 $\quad$ **return** d.
7 **else**
8 $\quad$ **return** ("Falla").

Este algoritmo se efectúa en tiempo $O(\log(n)^3)$ ya que el mayor tiempo de ejecución estará en la instrucción 4 en el calculo del máximo común divisor. El algoritmo $p-1$ suele ser exitoso en el afortunado caso en que n tiene un divisor primo p para el cual $p-1$ no tiene grandes factores primos. Supongamos que $(p-1)|k$ y tal que $p \nmid a$ y que $|\mathbb{Z}_p^*| = p-1$, se tiene que $a^k \equiv 1 \pmod{p}$, así $p|\gcd(a_k - 1, n)$. En muchos casos, tenemos que $p = \gcd(a_k - 1, n)$, por lo que el método encuentra un factor no trivial de n. En el peor caso, donde $(p-1)/2$ es primo, el algoritmo $p-1$ no es mejor que el método de la división de prueba. Como el grupo tiene orden fijo $p-1$, no hay nada que hacer excepto intentar con un algoritmo diferente.

Existe un método similar a $p-1$, llamado $p+1$, propuesto por H. C. Williams en 1982. Es adecuado para el caso en que n tiene una factor primo p para el cual $p+1$ no tiene grandes factores primos. Por lo tanto, el ataque $p-1$ como $p+1$ será un ataque útil para el criptosistema RSA si p o $q \pmod{n}$ tienen la propiedad que $p+1$ o $q+1$ es un número suave.

Para finalizar indicamos que **el algoritmo Quadratic Sieve**, inventado por Pomerance en 1981 y publicado por primera vez en 1982 [Yan, 2007], pertenece junto con **el algoritmo Number Field Sieve** y **el algoritmo de las fracciones continuas**, a una amplia gama de algoritmos de factorización, denominados **algoritmos de cálculo de índice de factorización**. Todos ellos hacen uso de la observación de que si tenemos dos enteros x y y tales que:

$$x^2 \equiv y^2 \pmod{n}, 0 < x < y < n, x \neq y, x + y \neq n;$$

entonces el $\gcd(x \pm y, n)$ son posibles factores no triviales de n, porque $n|(x+y)(x-y)$ pero $n \nmid (x+y)$ y $n \nmid (x-y)$. ¿Cómo encontrar x, y tal que la congruencia anterior se satisface?. Esta es la tarea principal del algoritmo de cálculo de índices; diferentes métodos utilizan diferentes técnicas para encontrar tales pares de números (x, y).

Se conjetura que estos algoritmos tienen un tiempo de ejecución [Yan, 2007]:

$$O(\exp\left((1+o(1)\sqrt{\log(n)\log(\log(n))}\right)) = O\left(n^{(1+o(1)\sqrt{(\log(\log(n)))/\log(n)}}\right).$$

2.3. Criptosistema de clave pública ElGamal

Definición 2.21 Consideremos el grupo cíclico $\mathbb{Z}_p^*$ de orden $p-1$, un elemento primitivo $\alpha \in \mathbb{Z}_p^*$ y otro elemento $\beta \in \mathbb{Z}_p^*$. El problema del logaritmo discreto (PLD) es el problema de determinar el entero $1 \leq x \leq p-1$ tal que:

$$\alpha^x \equiv \beta \pmod{p}.$$

Definición 2.22 Para el problema generalizado del logaritmo discreto tomamos un grupo cíclico finito G con la operación $\circ$ y cardinal n. Consideremos un elemento primitivo $\alpha \in G$ y otro elemento $\beta \in G$. El problema del logaritmo discreto consiste en encontrar el entero x, donde $1 \leq x \leq n$, tal que:

$$\beta = \underbrace{\alpha \circ \alpha \circ \cdots \circ \alpha}_{x \text{ veces}} = \alpha^x.$$

A continuación describimos el criptosistema ElGamal cuya seguridad depende de la complejidad del PLD.

Criptosistema 2.23 Sea p un número primo tal que el problema del logaritmo discreto en $(\mathbb{Z}_p^*, .)$ no es factible y sea $\alpha \in \mathbb{Z}_p^*$ un elemento primitivo. Con esto, $\mathcal{P} = \mathbb{Z}_p^*, \mathcal{C} = \mathbb{Z}_p^* \times \mathbb{Z}_p^*$ y definimos

$$\mathcal{K} = \{(p, \alpha, a, \beta) : \beta \equiv \alpha^a \pmod{p}\}.$$

Los valores p, α y β son la clave pública y a es la clave privada.

Para $K = (p, \alpha, a, \beta)$ y para un número aleatorio $k \in \mathbb{Z}_{p-1}$, definimos

$$e_k(x, k) = (y_1, y_2),$$

donde

$$y_1 = \alpha^k \pmod{p}$$

y

$$y_2 = x\beta^k \pmod{p}.$$

Para descifrar $(y_1, y_2) \in \mathbb{Z}_p^* \times \mathbb{Z}_p^*$, definimos

$$d_K(y_1, y_2) = y_2(y_1^a)^{-1} \pmod{p}.$$

Para generar la clave con seguridad, hemos de selecionar un número primo p cualquiera pero tal que el logaritmo discreto sea intratable. Esto es posible pues se trata de un problema NP-HARD. A continuación se han de seleccionar dos números aleatorios y sean estos g, a de tal forma que g es el generador del grupo cíclico Z_p^* y $a \in \{0, \ldots, p-1\}$, que será la clave privada.

Finalmente, se ha de calcular $K = g^a \pmod{p}$ y la clave pública será (g, p, K) , donde a es un valor secreto.

2.3.1. Ataques al criptosistema El Gamal

Como en todos los sistemas de cifrado de clave púbica, existen algunos ataques conocidos que hay que tener en cuenta para evitar la ruptura del sistema. Resumimos brevemente algunos de ellos.

El ataque Baby step - Giant step

Si trabajamos módulo m y escribimos $m = \lfloor \sqrt{n} \rfloor$, este algoritmo se basa en la observación de que si $x = \log_a(b)$, entonces, se puede escribir de forma única $x = i + jm$ [Stinson, 2005], donde $0 \leq i, j \leq m$. Este algoritmo (ver en capítulo 5) requiere una tabla cuyo tiempo de cálculo es del orden $O(m)$. Usando un algoritmo de ordenamiento como **el algoritmo Quicksort** que se ejecuta en tiempo $O(m \log(m))$ [Cormen et al., 2009], se proporciona un algoritmo para calcular logaritmos discretos que

se puede efectuar en $O(\sqrt{n}\log(n))$. Finalmente dada una entrada, se puede utilizar un algoritmo de búsqueda para determinar el logaritmo discreto correspondiente. Aunque este algoritmo funciona en grupos arbitrarios, si el orden de un grupo es mayor que 10^{40}, no es factible [Yan, 2007].

Ataque ρ y λ de Pollard

El ataque Baby step - Giant step es un tipo de método que forma parte de los denominados de raíz cuadrada para calcular logaritmos discretos. En 1978 Pollard dio otros dos tipos de métodos de este tipo, **el método ρ** y **el método λ** para obtener logaritmos discretos (ver capítulo 5). Los métodos de Pollard son probabilísticos pero eliminan la necesidad de precomputar tablas con logaritmos discretos como el ataque Baby step - Giant step. Los algoritmos de Pollard requieren $O(n)$ operaciones y por lo tanto, no son factibles cuando el orden del grupo G es mayor que 10^{40}.

Ataque de Silver–Pohlig–Hellman

Tambien en 1978 Pohlig y Hellman propusieron el algoritmo Silver-Pohlig-Hellman para calcular logaritmos discretos sobre campos finitos $GF(q) : \mathbb{F}_q$ con $O(\sqrt{p})$ operaciones y una cantidad similar de almacenamiento. Aquí p es el mayor factor primo de $q-1$. Ellos mostraron que si

$$q-1 = \prod_{i=1}^{k} p_i^{\alpha_i},$$

donde los p_i son primos distintos y los α_i son números naturales, y además, si $r_1, r_2, \ldots, r_k$ son números reales con $0 \leq r_i \leq 1$ entonces los logaritmos sobre $\mathbb{F}_q$ pueden ser calculados en

$$O\left(\sum_{i=0}^{k}(\log(q) + p_i^{1-r_i}(1+\log(p_i^{r_i})))\right)$$

operaciones en el campo, usando

$$O\left(\log(q)\sum_{i=1}^{k}(1+p_i^{r_i})\right)$$

bits de memoria, requiriendo cálculos pre computados con

$$O\left(\sum_{i=1}^{k} p_i^{r_i}\log(p_i^{r_i}) + \log(q)\right)$$

operaciones. Este algoritmo es muy eficiente si q es suave [Yan, 2007], esto es que todos los factores primos de $q-1$ son pequeños.

Finalmente describiendo un último ataque.

Ataque del cálculo de índice

Se trata de un algoritmo que explota la representación de los elementos del grupo como producto de elementos de un subconjunto pequeño. Sea $A = \langle g \rangle$ el grupo base, $\#A = n$ y $B = \{p_1, p_2, \ldots p_r\} \subseteq A$ lo que se denomina una base. Entonces:

1. Se buscan identidades del tipo

$$\prod_{l=1}^{r} p_j^{a_j} = g^t, t \in \mathbb{Z}^+,$$

o lo que es equivalente

$$\sum_{j=1}^{r} a_j \mathrm{ind}_g(p_j) \equiv t \pmod{n},$$

donde si $x = g^s$, $\mathrm{ind}_g(x) = s$.

2. Una vez obtenidas suficientes identidades se determinan los $\text{ind}_g(p_j)$ resolviendo un sistema de ecuaciones.

   Estas dos etapas primarias constituyen un proceso de pre-computación realizada a priori y cuyos datos se almacenan en una tabla.

3. Dado $a \in A$, para calcular $\text{ind}_g(a)$, que equivale a resolver el problema del logaritmo discreto, se buscan relaciones de la forma

$$\prod_{j=1}^{r} p_j^{e_j} = ag^e, e \in \mathbb{Z},$$

   con esto

$$\text{ind}_g(a) = \sum_{i=1}^{r} e_j \text{ind}(p_j) - e.$$

La eficacia del método depende fuertemente de que se pueda determinar un conjunto adecuado B para que el existan relaciones de la etapa 2 del método. Según [Yan, 2007] el tiempo de ejecución es

$$O\left(\exp\left(c\sqrt{\log(n)\log(\log(n)).}\right)\right).$$

2.3.2. Criptosistemas basados en curvas elípticas

Definición 2.24 Sean $a, b \in \mathbb{R}$ tales que $4a^3 + 27b^2 \neq 0$. Una curva elíptica no singular es el conjunto E de soluciones $(x, y) \in \mathbb{R} \times \mathbb{R}$ de la ecuación

$$y^2 = x^3 + ax + b,$$

junto con un punto especial $\mathcal{O}$ llamado el punto en el infinito. Si $4a^3+27b^2 = 0$ entonces la correspondiente curva elíptica se denomina curva elíptica singular.

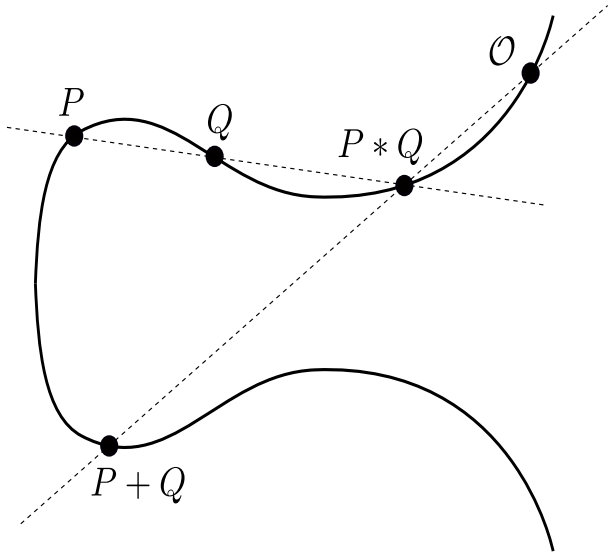
Sea E una curva elíptica no singular, a continuación definimos una operación + tal que $(E, +)$ es un grupo abeliano donde el punto $\mathcal{O}$ es el elemento identidad. Si $P, Q \in E$ tales que $P = (x_1, y_1)$ y $Q = (x_2, y_2)$ definimos $P + Q$ así:

- Si $x_1 \neq x_2$, entonces $(x_1, y_1) + (x_2, y_2) = (x_3, y_3)$, donde

$$\begin{aligned} x_3 &= \lambda^2 - x_1 - x_2; \\ y_3 &= \lambda(x_1 - x_3) - y_1, \text{ siendo} \\ \lambda &= \frac{y_2 - y_1}{x_2 - x_1}. \end{aligned}$$

- Si $x_1 = x_2$ y $y_1 = -y_2$, entonces $(x_1, y_1) + (x_2, y_2) = \mathcal{O}$.

- Si $x_1 = x_2$ y $y_1 = y_2$, entonces los valores de x_3 y y_3 son iguales al primer caso excepto que $\lambda = \dfrac{3x_1^2 + a}{2y_1}$.



$P + Q$ es el punto intersección de la curva con la recta que pasa por los puntos $P * Q$ y el punto del infinito.

Para conocer la estructura de una curva elíptica es necesario determinar el número de puntos racionales que posee. Es decir, si se define sobre $\mathbb{Z}_p$ el número de valores en $\mathbb{Z}_p \times \mathbb{Z}_p$ de la curva. Determinar este número por un un método efectivo, en $\mathbb{R}$ o para p grande, sigue siendo una cuestión abierta [Gómez and Tena, 1997]. Nuestra definición de curva elíptica puede extenderse a cualquier cuerpo finito $\mathbb{F}_q$ de característica $\neq 2, 3$ y según el teorema de Hasse-Weil

$$|\#E(\mathbb{F}_q) - (q+1)| \leq 2\sqrt{q},$$

por lo que toda curva elíptica sobre $\mathbb{F}_q$ tiene cardinal $q + 1 + t$ con $|t| \leq 2\sqrt{q}$.

El logaritmo discreto elíptico

Anteriormente, se ha examinado el criptosistema de ElGamal, basado en el problema del logaritmo discreto sobre el grupo multiplicativo de un cuerpo finito $\mathbb{F}_q$, sin embargo el problema puede plantearse sobre cualquier grupo abeliano finito [Gómez and Tena, 1997]. Se exigen a tal grupo las siguientes condiciones:

1. El grupo ha de ser cíclico.
2. Debe de disponer de un algoritmo eficiente para la multiplicación de sus elementos.
3. El orden del grupo ha de ser conocido.

Aunque en general, el problema del logaritmo discreto se considera intratable, su dificultad puede variar según el grupo en el que se considera. Así se ha visto como métodos como el cálculo del índice aplicado a algunos grupos concretos como los de tipo $\mathbb{F}_{2^m}$, han puesto en duda la seguridad de los criptosistemas basados en ellos. Estos inconvenientes conducen a la búsqueda de otros candidatos como grupo base, una propuesta que parece buena es el uso del grupo $E(\mathbb{F}_q)$ de puntos en una curva elíptica sobre un cuerpo finito $\mathbb{F}_q$.

Tomando en cuenta las 3 condiciones antes mencionadas y lo dicho sobre curvas elípticas, el grupo de puntos de una curva elíptica sobre $\mathbb{F}_q$, cumple con ellas, y esto lleva a la creación de una variante del criptosistema de ElGamal basado en curvas elípticas.

Este esquema utiliza unos algoritmos llamados P.COMPRESS y P.DECOMPRESS que se describen en el capítulo 5 para el cuerpo finito $\mathbb{F}_q = \mathbb{Z}_p$. Indicamos el esquema a continuación.

Criptosistema 2.25 Simplified Elliptic Curve Integrated Encrypted Scheme. Sea E una curva elíptica definida sobre $\mathbb{Z}_p$ ($p > 3$ primo) tal que contiene un subgrupo cíclico $H = \langle P \rangle$ de orden primo n en el cual el problema del logaritmo discreto es NP-HARD.
Sea $\mathcal{P} = \mathbb{Z}_p^*, \mathcal{C} = (\mathbb{Z}_p \times \mathbb{Z}_2) \times \mathbb{Z}_p^*$. Se define

$$\mathcal{K} = \{(E, P, m, Q, n) : Q = mP\}.$$

Los valores **P, Q y n son la clave pública**, con $m \in \mathbb{Z}_n^*$ como la clave privada. Para $K = (E, P, m, Q, n)$, para un número aleatorio (secreto) $k \in \mathbb{Z}_n^*$ y para $z \in \mathbb{Z}_p^*$ se define:

$$e_K(x, k) = (P.COMPRESS(kP), xx_0 \pmod p),$$

donde $kQ = (x_0, y_0)$ y $x_0 \neq 0$.

Para un texto cifrado $y = (y_1, y_2)$, donde $y_1 \in \mathbb{Z}_p \times \mathbb{Z}_2$ con $y_2 \in \mathbb{Z}_p^*$, se define

$$d_K(y) = y_2(x_0)^{-1} \pmod p,$$

donde $(x_0, y_0) = m$ P.DECOMPRESS(y_1).

Aunque el problema del logaritmo discreto parece más seguro en el caso elíptico que en el caso clásico sobre $\mathbb{F}_q$, son necesarias algunas precauciones tales como que el orden del grupo sea suficientemente grande y si además el cardinal del grupo de puntos de la curva tenga todos sus factores primos pequeños, en otro caso puede ser atacado con el método de Silver-Pohlig-Hellman.

2.4. Criptografía post-cuántica

En los últimos años ha crecido el interés en el tema de la computación cuántica: máquinas que explotan los fenómenos mecánico-cuánticos para lograr resolver problemas matemáticos que son difíciles de resolver por las computadoras actuales. El Instituto Nacional de Estándares y Tecnología (NIST) en la presentación del proyecto post-quatum predijo que dentro de los próximos 20 años se construirán computadoras cuánticas que podrán romper todos los sistemas de clave pública actualmente en uso.

Podríamos decir que el inicio de esta criptografía se produjo en 1997, Peter W. Shor [Shor, 1997] presenta un algoritmo cuántico para factorizar enteros sobre $\mathbb{Z}$ y para calcular logaritmos en el grupo multiplicativo $\mathbb{F}_q$ que se ejecuta en tiempo polinomial esto hizo que la comunidad científica empezara a tomar interés en criptografía basada en métodos cuánticos. Una vez la computadora cuántica sea una realidad , todas los sistemas de seguridad que utilizan los esquemas de la criptografía actual, basada en la infactibilidad de resolver ciertos problemas de la teoría de números será insegura, siendo preciso generar nuevos criptosistemas.

Por todo lo anterior, el NIST inició el proyecto de criptografía post-cuántica, que tiene como objetivo definir nuevos estándares para la criptografía y fijó el plazo para la presentación de algoritmos criptográficos de clave pública para noviembre de 2017 (véase `http://csrc.nist.gov/groups/ST/post-quantum-crypto/index.html`). El NIST ha exigido que los algoritmos sean fáciles de implementar, que tengan parámetros ajustables, que se puedan implementar en varias plataformas y aplicaciones, que sean paralelizables y que sean además resistente a los ataques clásicos en el campo criptográfico.

Actualmente el proyecto se encuentra en la primer etapa después de haber recibo 69 candidatos donde al menos uno de ellos se convertirá en el siguiente estándar de encriptación resistente a computadora cuántica.

Capítulo 3

Códigos lineales de evaluación

3.1. Introducción

Los códigos lineales de evaluación pueden verse como una versión local y más grande de los códigos geométrico algebraicos. Se hará un desarrollo de ellos sin recurrir a la teoría propiamente dicha de los objetos de interés de la geometría algebraica, para ello se introduce el concepto de función de orden y a partir de este concepto se definirán los códigos de evaluación y sus propiedades: distancia mínima, dimensión y algoritmos de decodificación.

La referencia que se ha tomado para este capítulo es [Hoholdt et al., 1998].

Si $\mathcal{P} = \{P_1, P_2, \ldots, P_n\}$ es un conjunto de puntos pertenecientes a un objeto geométrico $\mathcal{X}$ (como, por ejemplo, una curvas algebraica) y si V es un espacio vectorial de funciones $f : \mathcal{X} \to \mathbf{F}_q$, se puede considerar la aplicación de evaluación en $\mathcal{P}$

$$ev_{\mathcal{P}} : V \to \mathbf{F}_q, ev_{\mathcal{P}}(f) = (f(P_1), f(P_2), \ldots, f(P_n)).$$

Si $ev_{\mathcal{P}}$ es lineal, su imagen es un código lineal sobre $\mathbf{F}_q$ de longitud n y sus palabras son los elementos $ev_{\mathcal{P}}(f), f \in V$. Diremos que este código es obtenido por evaluación en $\mathcal{P}$ de las funciones de V y sus parámetros pueden deducirse de las propiedades de V.

3.2. Funciones de orden

Definición 3.1 Una relación $\prec$ sobre un conjunto S se denomina:

- **reflexiva**: Si $a \prec a, \forall a \in S$.
- **simétrica**: Si $a \prec b$ entonces $b \prec a, \forall a \in S$.
- **antisimétrica**: Si se cumple que $a \prec b$ y $b \prec a$ entonces $a = b$.
- **transitiva**: Si $a \prec b$ y $b \prec c$ entonces $a \prec c$.

Definición 3.2 Una relación de orden parcial es una relación que es reflexiva, antisimétrica y transitiva.

Definición 3.3 Una relación de orden total es un orden parcial donde se cumple **la ley de tricotomía**, esto es, para cualquier $a, b \in S$ se cumple que $a \prec b$, $b \prec a$ o $a = b$.

Definición 3.4 Una $\mathbb{F}$-álgebra es un anillo conmutativo con uno, que contiene al campo $\mathbb{F}$ como un subanillo unitario.

Definición 3.5 Sea $R = \mathbb{F}[X_1, X_2, \ldots, X_n]$ el anillo de polinomios con coeficientes en $\mathbb{F}$ en n variables y $\prec$ un orden total en el conjunto de monomios en las variables $X_1, X_2, \ldots, X_n$ tal que para todo monomio M_1, M_2 se cumple que

$$\begin{array}{rcl} \text{Si } M \neq 1 & \Rightarrow & 1 \prec M \\ \text{Si } M_1 \prec M_2 & \Rightarrow & MM_1 \prec MM_2 \end{array}$$

Entonces $\prec$ se denomina orden admisible sobre los monomios.

Utilizando la notación multi-índice se tiene que $X^\alpha = \prod_{i=1}^{m} X_i^{\alpha_i}$, si $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_m)$ entonces el grado de un monomio X^α, se define como:

$$\text{grad}(X^\alpha) = \text{grad}(\alpha) = \sum_{i=1}^{m} \alpha_i.$$

Definición 3.6 Dados los monomios X^α y X^β, $X^\alpha \prec_L X^\beta$ si y solo si $x^\alpha = x^\beta$ (si no pasa esto no es reflexiva) ó $\alpha_1 = \beta_1, \alpha_2 = \beta_2, \ldots, \alpha_{l-1} = \beta_{l-1}$ y $\alpha_l < \beta_l$ para algún $l, 1 \leq l \leq m$.

El orden $\prec_L$ se llama lexicográfico y es un orden admisible tal que si $m \geq 2$ no es isomórfico con los enteros positivos con el orden usual.

Definición 3.7 Dados los monomios X^α y X^β, decimos que $X^\alpha \prec_D X^\beta$ si y solo si

$$\text{grad}(X^\alpha) < \text{grad}(X^\beta) \text{ o } \text{grad}(X^\alpha) = \text{grad}(X^\beta) \text{ con } X^\alpha \prec_L X^\beta.$$

El orden $\prec_D$ se llama graduado lexicográfico, es un orden admisible el cual es isomórfico con los números enteros positivos con el orden usual. Sea $\prec$ un orden admisible isomórfico con los números enteros positivos con el orden usual y sean $f_1, f_2, \ldots,$ la enumeración de los monomios tales que $f_i \prec f_{i+1}$, para todo i; los monomios forman una base de $\mathbb{F}[X_1, X_2, \ldots, F_n]$ sobre $\mathbb{F}$, entonces cada polinómio no nulo puede ser escrito de forma única como

$$f = \sum_{i=1}^{j} \lambda_i f_i,$$

donde $\lambda_i \in \mathbb{F}$, para todo i, además, $\lambda_j \neq 0$.

Definición 3.8 Sea R una $\mathbb{F}$-álgebra y supongamos $-\infty < n, \forall n \in \mathbb{N}_0$. Una función $\rho : R \to \mathbb{N} \cup \{-\infty\}$ se llama función orden si satisface

1. $\rho(f) = 0$ si y solo si $f = 0$.
2. $\rho(\lambda f) = \rho(f)$ para todo $\lambda \in \mathbb{F}$ no nulo.
3. $\rho(f + g) \leq \text{máx}\{\rho(f), \rho(g)\}$ y la igualdad se cumple cuando $\rho(f) < \rho(g)$.
4. Si $\rho(f) < \rho(g)$ y $h \neq 0$ entonces $\rho(fh) < \rho(gh)$.
5. Si $\rho(f) = \rho(g)$, entonces existe un $\lambda \in \mathbb{F}$ no nulo tal que $\rho(f - \lambda g) < \rho(g)$.

para todo $f, g, h \in R$.

Con la notación del párrafo anterior a la definición 3.8, la función $\rho : \mathbb{F}[X_1, X_2, \ldots, X_m] \to \mathbb{N}_0 \cup \{-\infty\}$ tal que $\rho(0) = -\infty$ y $\rho(f) = j - 1$ donde j es el menor entero positivo tal que f puede ser escrito como una combinación lineal de los primeros j monomios es una función orden.

Definición 3.9 Una función peso es una función orden sobre R que además satisface que

6. $\rho(gf) = \rho(f) + \rho(g)$.

para todo $f, g, h \in R$. Aquí se conviene que $-\infty + n = -\infty, \forall n \in \mathbb{N}_0$.

Si ρ es una función peso tal que $\rho(f)$ es divisible por un entero $d > 1$, para todo $f \in R$ entonces $\rho(f)/d$ es también una función peso.

Definición 3.10 Una función grado es una función $R \to \mathbb{N}_o$ que satisface las condiciones 1,2,3 y 6.

Enunciamos a continuación unas propiedades sobre las funciones anteriores, las pruebas pueden encontrarse en [Hoholdt et al., 1998].

Lema 3.11 Sea ρ una función orden sobre R, entonces:

1. Si $\rho(f) = \rho(g)$, entonces $\rho(fh) = \rho(gh)$, para todo $h \in R$.
2. Si $f \in R$ y $f \neq 0$ entonces $\rho(1) \leq \rho(f)$.
3. $\mathbb{F} = \{f \in R | \rho(f) \leq \rho(1)\}$.
4. Si $\rho(f) = \rho(g)$, entonces existe un único $\lambda \in \mathbb{F}$ no nulo tal que $\rho(f - \lambda g) < \rho(g)$.

Proposición 3.12 Si existe una función orden sobre R, entonces R es un dominio integral.

Prueba: Supongamos que $fg = 0$ para $f, g \in R$ no nulos y asúmase que $\rho(f) \leq \rho(g)$, entonces $\rho(f^2) \leq \rho(fg) = \rho(0) = -\infty$, esto es que $\rho(f^2) = -\infty$ y $f^2 = 0$, ahora $f \neq 0$, siendo que $\rho(1) \leq \rho(f)$ por el Lema 3.11, entonces $\rho(f) \leq \rho(f^2) = \rho(0) = -\infty$ pero esto es una contradicción, por tanto R no tiene divisores de cero no nulos. □

Proposición 3.13 Sea R una $\mathbb{F}$-álgebra con función orden ρ y $R \neq \mathbb{F}$, entonces existe una base $\{f_i | i \in \mathbb{N}\}$ de R sobre $\mathbb{F}$ tal que $\rho(f_i) < \rho(f_{i+1})$ para todo i. Además, si i es el entero positivo más pequeño tal que f puede ser escrito como una combinación lineal de los primeros i elementos de la base, entonces $\rho(f) = \rho(f_i)$. Finalmente sea $l(i, j)$ un entero l tal que $\rho(f_i f_j) = \rho(f_l)$, entonces $l(i, j) < l(i + 1, j)$ para todo i, j y si además ρ es una función peso entonces $\rho_{l(i,j)} = \rho_i + \rho_j$.

Teorema 3.14 Sea R una $\mathbb{F}$-álgebra y $\{f_i | i \in \mathbb{N}\}$ una base de R como espacio vectorial sobre $\mathbb{F}$, donde $f_1 = 1$. Denotamos por L_i el espacio generado por $f_1, f_2, \ldots, f_i$. Sea $l(i, j)$ el entero positivo mas pequeño tal que $f_i f_j \in L_l$ y supongamos que $l(i, j) < l(i+1, j), \forall i, j \in \mathbb{N}$. Sea $(\rho_i | i \in \mathbb{N})$ un secuencia estrictamente creciente de enteros no negativos, se define $\rho(0) = -\infty$ y $\rho(f) = \rho_i$. Si i es el entero positivo más pequeño tal que $f \in L_i$ entonces ρ es una función orden sobre R y si además $\rho_{l(i,j)} = \rho_i + \rho_j$ entonces ρ es una función peso.

3.3. Códigos de evaluación

En esta subsección consideramos un $\mathbb{F}_q$-álgebra R con una función orden ρ. Sea $(f_i | i \in \mathbb{N})$ una base de R sobre $\mathbb{F}_q$ tal que $\rho(f_i) < \rho(f_{i+1})$, para todo $i \in \mathbb{N}$. Denotamos L_l espacio generado por $f_1, f_2, \ldots, f_l$. Finalmente, escribimos $l(i, j)$ el entero positivo más pequeño l tal que $f_i f_j \in L_l$.

La multiplicación coordinada sobre $\mathbb{F}_q^n$ se define $a * b = (a_1 b_1, a_2 b_2, \ldots, a_n b_n)$, donde $a = (a_1, a_2, \ldots, a_n)$ y $b = (b_1, b_2, \ldots b_n)$. El espacio vectorial $\mathbb{F}_q^n$ con la multiplicación $*$ se convierte en un anillo conmutativo con unidad $(1, 1, \ldots, 1)$. Se identifica el subanillo unitario $\{(\lambda, \lambda, \ldots, \lambda) | \lambda \in \mathbb{F}_q\}$ con $\mathbb{F}_q$. En esta forma $\mathbb{F}_q^n$ es una $\mathbb{F}_q$-álgebra.

Definición 3.15 Un mapeo $\varphi : R \to \mathbb{F}_q^n$ se dice que es un morfismo de $\mathbb{F}_q$-álgebras si φ es $\mathbb{F}_q$-lineal y $\varphi(fg) = \varphi(f) * \varphi(g)$.

Definición 3.16 Con la notación anterior, escribimos $h_i = \varphi(f_i)$. El código de evaluación E_l y su código dual C_l se definen como

$$E_l = \varphi(L_l) = \langle h_1, h_2, \ldots, h_l \rangle$$

$$C_l = \{c \in \mathbb{F}_q^n | c.h_i = 0, \text{ para todo } i \leq l\}$$

La secuencia de códigos $(E_l|l \in \mathbb{N})$ es creciente con respecto a la inclusión. Todos estos códigos son subespacios vectoriales de $\mathbb{F}_q^n$, por ello existe un N tal que $E_l = E_N$ para todo $l \geq N$. El código E_N es la imagen de R sobre φ y si se consideran solo aquellos morfismos φ que son sobreyectivos, entonces $E_l = \mathbb{F}_q^n$ y $C_l = 0$, para todo $l \geq N$.

Definición 3.17 Sea $\mathcal{P}$ un conjunto formado por n puntos distintos $P_1, P_2, \ldots, P_n$ de $\mathbb{F}_q^n$. Considérese el mapeo de evaluación

$$ev_{\mathcal{P}} : R[X_1, X_2, \ldots, X_m] \to \mathbb{F}_q^n$$

definido como $ev_{\mathcal{P}}(f) = (f(P_1), f(P_2), \ldots, f(P_n))$, este es un morfismo de $\mathbb{F}_q$-álgebras de R a $\mathbf{F}_q^n$, ya que $fg(P) = f(P)g(P)$ para todo par de polinomios f, g y todo los puntos P de $\mathbb{F}_q^M$.

Lema 3.18 El mapeo $ev_{\mathcal{P}}$ es sobreyectivo.

Prueba: Sea $P_j = (x_{j1}, x_{j2}, \ldots, x_{jm})$. Si $A_{il} = \{x_{jl}|j = 1, 2, \ldots, n\} \setminus \{x_{il}\}$. Se definen los polinomios

$$G_i = \prod_{l=1}^{m} \prod_{x \in A_{il}} (X_l - x).$$

Entonces $G_i(P_j) = 0$, para todo $i \neq j$ y además $G_i(P_i) \neq 0$, porque los puntos $P_1, P_2, \ldots, P_n$ son distintos. El polinomio $G_i/G_i(P_i)$ mapea vía $ev_{\mathcal{P}}$ en el i-ésimo elemento de la base canónica de $\mathbb{F}_q^n$. Por lo tanto $ev_{\mathcal{P}}$ es sobreyectiva.

Se considera ahora un ideal I en el anillo $\mathbb{F}[X_1, X_2, \ldots, X_m]$ y $P_1, P_2, \ldots, P_n$ puntos en el conjunto de ceros de I con coordenadas en $\mathbb{F}$. Entonces el mapeo de evaluación induce un mapeo lineal bien definido

$$ev_{\mathcal{P}} : \mathbb{F}[X_1, X_2, \ldots, X_m] \setminus I \to \mathbb{F}_q^n,$$

el cual es un morfismo sobreyectivo de $\mathbb{F}$-álgebras. □

Sabemos que existe un entero N tal que $E_l = \mathbb{F}_q^n$ para todo $l > N$. Con la notación anterior se considera H de tamaño $N \times n$ donde h_i es su i-ésima columna para $1 \leq i \leq N$.

Definición 3.19 Para $y \in \mathbb{F}_q^n$, consideremos los síndromes

$$s_i(y) = y.h_i \text{ y } s_{ij}(y) = y.(h_i * h_j).$$

Por definición $S(y) = (s_{ij}(y)|1 \leq i, j \leq N)$ es **la matriz de síndromes de y.**

Lema 3.20 Sea $y \in \mathbb{F}_q^n$ y $D(y)$ la matriz diagonal con y en su diagonal. Entonces:

$$S(y) = HD(y)H^T$$

y

$$\text{rang}(s(Y)) = \text{wt}(y).$$

Prueba: La matriz de síndromes $S(y)$ es igual a $HD(y)H^T$, por lo que

$$s_{ij}(y) = y.(h_i * h_j) = \sum_l y_l h_{il} h_{jl},$$

donde h_{il} es la l-ésima entrada de h_i. El rango de la matriz diagonal $D(y)$ es igual al número de entradas distintas de cero en y, que es $\text{wt}(y)$. Las columnas de H generan $\mathbb{F}_q^n$, por lo que $E_N = \mathbb{F}_q^n$. Las matrices H y H^T tienen ambas rango completo n, por tanto $\text{rang}(S(y)) = \text{rang}(D(y)) = \text{wt}(y)$. □

Continuando con la notación anterior podemos dar la siguiente definición del conjunto N_l.

Definición 3.21 Sea $l \in \mathbb{N}_0$, se define

$$N_l = \{(i, j) \in \mathbb{N}^2 | l(i, j) = l + 1\}.$$

Lema 3.22 Se cumplen las siguientes propiedades:

1. $y \in C_l$ y $l(i,j) \leq l$, entonces $s_{ij}(y) = 0$.
2. $y \in C_l \setminus C_{l+1}$ y $l(i,j) = l+1$, entonces $s_{ij}(y) \neq 0$.

Prueba:
1. Sea $y \in C_l$, si $l(i,j) \leq l$ entonces $f_i f_j \in L_l$, por tanto $h_i * h_j = \rho(f_i f_j)$ es un elemento de $\rho(L_l)$, el cual es el dual de C_l, $s_{ij}(y) = y.(h_i * h_j) = 0$.
2. Sea $y \in C_l \setminus C_{l+1}$, si $l(i,j) = l+1$, entonces $f_i f_j \in L_{l+1} \setminus L_l$, entonces $f_i f_j \equiv \mu f_{l+1}$ (mód L_l) para algún $\mu \in \mathbb{F}_q$ distinto de cero. Por tanto $h_i * h_j \equiv \mu h_{l+1}$ (mód $\rho(L_l)$). Como $y \notin C_{l+1}$, $s_{l+1}(y) \neq 0$. Por ello $s_{ij}(y) \neq 0$. □

En lo que sigue v_l denota el número de elementos de N_l.

Lema 3.23 Escribamos por comodidad $t = v_l$ y supongamos que $(i_1, j_1), (i_2, j_2), \ldots, (i_t, j_t)$ es una enumeración de los elementos de N_l en orden creciente con respecto al orden lexicográfico en $\mathbb{N}^2$. Entonces $i_1 < i_2 < \ldots < i_t$ y $j_t < j_{t-1}, \ldots < j_1$. Si además $y \in C_l \setminus C_{l+1}$, entonces

$$s_{i_u j_u} = \begin{cases} 0 & \text{si} \quad u < v, \\ \neq 0 & \text{si} \quad u = v. \end{cases}$$

Prueba: La secuencia $(i_1, j_1), (i_2, j_2), \ldots, (i_t, j_t)$ está ordenada de tal forma que $i_1 \leq \cdots \leq i_t$ y $j_u < j_{u+1}$, si $i_u = i_{u+1}$. Además, si $j_u = j_{u+1}$, entonces

$$l + 1 = l(i_u, j_u) < l(i_u, j_{u+1}) = l(i_{u+1}, j_{u+1}) = l + 1,$$

cual es una contradicción. Por lo tanto la secuencia $i_1, i_2, \ldots, i_t$ es estrictamente creciente. De igual forma se concluye que $j_{u+1} < j_u$, para todo $u < t$.

Sea $y \in C_l$, si $u < v$, entonces $l(i_u, j_v) < l(i_v, j_v) = l + 1$ y por el Lema 3.22 $s_{i_u j_v}(y) = 0$. Además, si $y \notin C_{l+1}$ y si $u = v$, entonces $l(i_u, j_v) = l + 1$ y por el Lema 3.22 $s_{i_u j_v}(y) \neq 0$.

De los lemas 3.22 y 3.23 podemos además concluir que si $y \in C_l \setminus C_{l+1}$, entonces $\mathrm{wt}(y) \geq v_l$.

Proposición 3.24 Si $y \in C_l \setminus C_{l+1}$, entonces $\mathrm{wt}(y) \geq v_l$

Prueba: Es consecuencia del Lema 3.20 y Lema 3.23.

Definición 3.25 Con la notación anterior, los valores

$$d(l) = \text{mín}\{v_m | m \geq l\} \text{ y}$$

$$d_\varphi(l) = \text{mín}\{v_m | m \geq l, C_m \neq C_{m+1}\}$$

son llamados cotas de orden, Además, si R es una álgebra afín de la forma $\mathbb{F}_q[X_1, X_2, \ldots, X_m]/I$ y φ es el mapeo de evaluación $ev_{\mathcal{P}}$ de un conjunto $\mathcal{P}$ de n puntos en $\mathbb{F}_q^m$, entonces d_φ se denota por $d_{\mathcal{P}}$.

Teorema 3.26 Las cotas de orden $d(l)$ y $d_\varphi(l)$ son cotas inferiores de la distancia mínima del código dual C_l

$$d(C_l) \geq d_\varphi(l) \geq d(l).$$

Prueba: El teorema es consecuencia directa de la Definición 3.25 y la Proposición 3.24.

- El conjunto N_l y los números $d(l)$ y v_l no dependen de la elección de la base $\{f_i | i \in \mathbb{N}\}$ ni de la elección del conjunto de puntos. Solo dependen de la función orden ρ.
- El número $d_{\mathcal{P}}$ depende de la función orden y de la elección del conjunto de los puntos, pero no de la elección de la base, por lo que si $\mathcal{P} \subseteq \mathcal{P}'$ entonces $d_{\mathcal{P}} \geq d_{\mathcal{P}'}$.

Sea $\mathbb{F}_0$ un sub-campo de $\mathbb{F}_q$, podemos definir el código

$$C_l^0 = \{c \in \mathbb{F}_0^n | c.h_i = 0, \forall i \leq l\},$$

entonces las cotas $d(l)$ y $d_\varphi(l)$ también serán distancias mínimas de C_l^0. Si se define

$$d_\varphi^0(l) = \text{mín}\{v_m | m \geq l, C_m^0 \neq C_{m+1}^0\},$$

entonces $d_\varphi^0(l) \geq d_\varphi(l)$ y estas son las cotas inferiores de la distancia mínima del código C_l^0. Todas estas cotas tienen la descomposición de la matriz de síndromes $S(y)$ en común, y patrones de ceros en la matriz dando información sobre los elementos distintos de cero de y.

Sean $\{a_1, a_2, \ldots, a_n\}, \{b_1, b_2, \ldots, b_n\}$ y $\{c_1, c_2, \ldots, c_n\}$ tres bases de $\mathbb{F}_q^n$. Denotamos por $\bar{E}_l$ el código generado por $c_1, c_2, \ldots, c_n$ y sea $\bar{C}_l$ su código dual. Póngase $\bar{l}(i,j)$ el entero positivo más pequeño l tal que $a_i * b_i \in \bar{E}_l$. El par (i,j) se denomina **bien portado** si $\bar{l}(i',j') < \bar{l}(i,j)$ para todo i', j' tales que $i' < i, j' < j$ y $(i',j') \neq (i,j)$. Para $l \in \{0, 1, \ldots, n-1\}$ se define

$$\bar{N}(l) = \{(i,j) | \bar{l}(i,j) = l+1 \text{ y } (i,j) \text{ son bien portados}\}.$$

Ahora escribimos $\bar{v}(l)$ el número de elementos de $\bar{N}(l), 0 \leq l \leq n-1$ y $\bar{v}(n) = n+1$, se define

$$\bar{d}(l) = \text{mín}\{\bar{v}(m) | l \leq m \leq n\}.$$

Entonces $\bar{d}(l)$ es una cota inferior de la distancia mínima de $\bar{C}_l$.

Si ahora consideramos la base $\{a_1, a_2, \ldots, a_n\}$ obtenida por borrar sucesivamente elementos superfluos de la secuencia $(\varphi(f_i) | i \in \mathbb{N})$ y tomamos las bases $\{b_1, b_2, \ldots, b_n\}$ y $\{c_1, c_2, \ldots, c_n\}$ obtenidas de forma similar a partir de $(\varphi(g_j) | j \in \mathbb{N})$ y $(\varphi(h_l) | l \in \mathbb{N})$, respectivamente. Si además escribimos k dimensión de C_l, $r = n - k$ y suponemos $C_l \neq C_{l+1}$, entonces $\bar{C}_r = C_l$ y $\bar{d}(r) \geq d\varphi(l)$.

Definición 3.27 Sea d un entero positivo, se define

$$\tilde{\text{C}}(d) = \{c \in \mathbb{F}_q^n | c.h_{l+1} = 0, \forall l \in \mathbb{N}_0 \text{ tal que } v_l < d\} \text{ y}$$

$$\tilde{\text{C}}_\varphi(d) = \{c \in \mathbb{F}_q^n | c.h_{l+1} = 0, \forall l \in \mathbb{N}_0 \text{ tal que } v_l < d \text{ y } C_l \neq C_{l+1}\}.$$

Entonces podemos deducir el siguiente resultado.

Proposición 3.28 La distancia mínima de los códigos $\tilde{\text{C}}(d)$ y $\tilde{\text{C}}\varphi(d)$ es al menos d.

Prueba: El código $\tilde{\text{C}}(d)$ esta contenido en $\tilde{\text{C}}\varphi(d)$. Sea y una palabra código distinta de cero de $\tilde{\text{C}}\varphi(d)$, si $d = 1$ entonces no hay nada que probar. Sea $d > 1$, entonces $v_0 = 1 < d$, por lo que $y.h_1 = 0$, el número N fue definido de tal forma que los elementos $h_1, h_2, \ldots, h_N$ generan $\mathbb{F}_q^n$, la palabra y es distinta de cero, entonces existe un entero positivo l tal que $y.h_{l+1} \neq 0$. Sea l el entero positivo más pequeño tal que $y.h_{l+1} \neq 0$, entonces $y \in C_l \setminus C_{l+1}$, por lo tanto $wt(y) \geq v_l$. Si $v_l < d$, entonces $y.h_{l+1} = 0$, ya que $y \in \tilde{\text{C}}_\varphi(d)$ y $C_l \neq C_{l+1}$, esto es una contradicción, por lo tanto $wt(t) \leq v_l \leq d$.

La siguiente definición permite dar más información sobre el anterior código.

Definición 3.29 Sea d un entero positivo, se definen los conjuntos

$$R(d) = \{l+1 | l \in \mathbb{N}_0, v_l < d\} \text{ y}$$

$$R_\varphi(d) = \{l+1 | l \in \mathbb{N}_0, v_l < d \text{ y } C_l \neq C_{l+1}\}.$$

Sean $r(d)$ y $r_\varphi(d)$ el número de elementos de $R(d)$ y $R_\varphi(d)$, respectivamente.

Los números anteriores permiten afirmar:

- El número $r(d)$ es el número de control de paridad que define $\tilde{\mathrm{C}}(d)$ y depende solo de la función orden.
- Este control de paridad puede ser dependiente, entonces la redundancia de $\tilde{\mathrm{C}}(d)$ es a lo mas $r(d)$.
- La dimensión de $\tilde{\mathrm{C}}(d)$ es al menos $n - r(d)$.
- El número $r_\varphi(d)$ es el número de control de paridad que define $\tilde{\mathrm{C}}_\varphi(d)$ y depende en la función orden y el mapeo ρ.
- En este caso y por definición, el control de paridad es independiente.
- la redundancia en $\tilde{\mathrm{C}}_\varphi(d)$ es igual a $r_\varphi(d)$.
- La dimensión de $\tilde{\mathrm{C}}_\varphi(d)$ es $n - r_\varphi(d)$.

Para acabar, indicamos que los códigos $\tilde{\mathrm{C}}(d)$ y $\tilde{\mathrm{C}}\varphi(d)$ tienen la propiedad de super código, esto es si $d = d(l)$ entonces $C_l \subseteq \tilde{\mathrm{C}}(d) \subseteq \tilde{\mathrm{C}}_\varphi(d)$, por lo que la distancia mínima de los códigos C_l, $\tilde{\mathrm{C}}(d)$ y $\tilde{\mathrm{C}}_\varphi(d)$ es al menos d, pero la de C_l podría ser más pequeña.

3.4. Funciones peso y semigrupos

3.4.1. Semigrupos y la distancia mínima

Empezamos indicando que la condición 6 de la definición de función peso ρ implica que el subconjunto $\Lambda = \{\rho(f) | f \in R, f \neq 0\}$ de los enteros no negativos $\mathbb{N}_0$ tiene la propiedad que $0 \in \Lambda$ y $x + y \in \Lambda$, para todo $x, y \in \Lambda$.

Esta propiedad es la que nos lleva a la definición de semigrupo.

Definición 3.30 Un subconjunto Λ de $\mathbb{N}_0$ se dice que es un semigrupo (numérico) si $0 \in \Lambda$ y para todo $x, y \in \Lambda$ también $x + y \in \Lambda$.

- Los elementos de $\mathbb{N}_0 \setminus \Lambda$ se denominan agujeros de Λ.
- Los elementos de Λ se denomina no agujeros de Λ.
- Si todos los elementos de Λ son divisibles por un entero $d > 1$, entonces existen infinitos agujeros en Λ.
- El número de agujeros de Λ se denota como $g = g(\Lambda)$.
- Si $g < \infty$ entonces existe un $n \in \Lambda$ tal que si $x \in \mathbb{N}_0$ y $x \geq n$, entonces $x \in \Lambda$.

Definición 3.31 El menor $n \in \Lambda$ tal que $\{x \in \mathbb{N}_0 | x \geq n\}$ está contenido en Λ, se llama conductor de Λ y se denota por $c = c(\Lambda)$. Consecuentemente $c - 1$ es el agujero más grande de Λ si $g > 0$.

Si Λ es un semigrupo con g agujeros y conductor c, entonces:

- $g = 0$ si y solo si $c = 0$.
- Si $g > 0$ entonces $c \geq g + 1$ y $\Lambda = \{x \in \mathbb{N}_0 | x \geq g + 1\} \cup \{0\}$ si y solo si $c = g + 1$.
- Existe exactamente un agujero si y solo si 1 es el único agujero.
- Si 2 no es un agujero, entonces $\{1, 3, 5, \ldots, 2g - 1\}$ es el conjunto de agujeros, en consecuencia $c = 2g$.

Definición 3.32 Los elementos de un semigrupo Λ pueden ser enumerados por la secuencia $(\rho_l | l \in \mathbb{N})$ tal que $\rho_l < \rho_{l+1}$, para todo l. El número de agujeros más pequeños que ρ_l es denotado como $g(l)$.

Lema 3.33 Sea Λ un semigrupo con un número finito de agujeros

1. Si $l \in \mathbb{N}$ entonces $g(l) = \rho_l - l + 1$.
2. Si $l \in \mathbb{N}$ entonces $\rho_l \leq l + g - 1$ y la igualdad se cumple si y solo si $\rho_l \geq c$.
3. Si $l > c - g$, entonces $\rho_l = l + g - 1$.
4. Si $l \leq c - g$ entonces $\rho_l < c - 1$.

Prueba:

1. El no agujero ρ_l es el $(\rho_l + 1)$-ésimo elemento de $\mathbb{N}_0$, entonces ρ_l es el $(\rho_l + 1 - g(l))$-ésimo elemento del semigrupo Λ, lo que prueba 1.
2. Este apartado se deduce de que $g(l) \leq g$ y $g(l) = g$ si y solo si $\rho_l \geq c$.
3. Para probar 3, observamos que c es el $(c + 1)$-ésimo elemento de $\mathbb{N}_0$ y todos los agujeros son estrictamente menores que c. Entonces c es el $(c+1-g)$-ésimo elemento de Λ, por lo que $c = \rho_{c+1-g}$. Ahora sea $l > c - g$, entonces $\rho_l \geq \rho_{c-g+1}$. Por tanto $\rho_l = l + g - 1$ por (2).
4. Para finalizar probamos 4, sea $l \leq c - g$, entonces $\rho_l \leq l + g - 1 \leq c - 1$, pero $c - 1$ es un agujero o es negativo, por lo que $\rho_l < c - 1$. □

Proposición 3.34 Por la notación anterior si el número de agujeros es finito, entonces

$$c \leq 2g$$

y $c = 2g$ si y solo si para cualquier entero no negativo s, si s es un agujero, entonces $c - 1 - s$ no es un agujero.

Prueba: Considérese un par de enteros no negativos (s, t) con $s + t = c - 1$, al menos uno de estos números es un agujero, ya que $c - 1$ es un agujero y la suma de dos no agujeros es un no agujero, pero hay c de tales pares, obteniendo la inecuación requerida.

La igualdad se obtiene si y solo si para cualquier par de enteros no negativos (s, t) con $s + t = c - 1$ exactamente uno de estos números es un no agujero y el otro es un agujero. □

Definición 3.35 Un semigrupo se llama simétrico si $c = 2g$.

Definición 3.36 Sea $A = \{a_1, a_2, \ldots, a_k\}$ un subconjunto de un semigrupo Λ. El semigrupo Λ se dice generado por A y se escribe $\Lambda = \langle A \rangle$, si para cualquier elemento $s \in \Lambda$ entonces existen $x_1, x_2, \ldots, x_k \in \mathbb{N}_0$ tal que $s = \sum_{i=1}^{k} x_i a_i$. Un conjunto A de generadores de Λ es minimal si Λ no es generado por ningún subconjunto propio de A.

En lo que sigue se tomaran por ciertos los siguientes hechos:

- Cada semigrupo tiene un conjunto finito de generadores.
- Cada conjunto de generadores contiene un conjunto minimal de generadores.
- El conjunto minimal de generadores es único.

Proposición 3.37 Sean $a, b \in \mathbb{N}$ tales que $\gcd(a, b) = 1$, el semigrupo generado por a y b es simétrico, tiene $ab - a - b$ como el más grande agujero, $(a - 1)(b - 1)$ como conductor y el número de agujeros es igual a $\dfrac{(a-1)(b-1)}{2}$.

Prueba: Cada entero tiene una única representación $m = ax + by$ donde x, y son enteros tales que $0 \leq y < b$ y $x < 0$, esto es debido a que $\gcd(a, b) = 1$. Como consecuencia cada agujero m tiene una única representación $m = bx + ay$ tal que $0 \leq y < b$ y $x < 0$ y cada no agujero m tiene una única representación $m = ax + by$ tal que $0 \leq y < b$ y $y \geq 0$.
Sea c el conductor del semigrupo Λ generado por $\langle a, b\rangle$. Está claro que $c - 1$ es el mayor agujero. Los números $ay \in \Lambda$, $y = 0, 1, \ldots, b - 1$ son un conjunto completo de los representantes de las clases b y además $ay - b$ es el mayor elemento en la clase de ay sin representación con coeficientes enteros no negativos. Por lo tanto $(b-1)a-b$ es el mayor agujero, el cual es es igual a $c-1$, entonces $c = (a-1)(b-1)$. Para ver que $\langle a, b\rangle$ es simétrico, se asume que s y t son ambos agujeros y $s + t = c - 1$. Se sabe que

$$\begin{array}{lcl} s = bx_1 + ay_1 & , & t = bx_2 + ay_2 \text{ donde} \\ 0 \leq y_1, y_2 < b & \text{y} & x_1, x_2 < 0. \end{array}$$

Entonces $c - 1 = ab - a - b = (x_1 + x_2)b + (y_1 + y_2)a$, por lo tanto

$$(-x_1 - x_2 - 1)b = (y_1 + y_2 - b + 1)a,$$

donde $0 \leq y_1 + y_2 \leq 2b - 2$ y $x_1 + x_2 \leq -2$. Teniendo en cuenta que el lado izquierdo de la última ecuación es estrictamente positivo y el lado derecho es estrictamente menor que ab, llegamos a una contradicción ya que $\gcd(a, b) = 1$.

Por lo tanto Λ es simétrico y $c = 2g$ por la Proposición 3.34, donde g es el número de agujeros, entonces se tiene $g = \dfrac{(a-1)(b-1)}{2}$. □

De la proposición anterior se deduce el siguiente colorario.

Colorario 3.38 Un semigrupo tiene un número finito de agujeros si y solo si el máximo común divisor de sus divisores es 1.

Prueba: Si el máximo común divisor de los elementos de un semigrupo Λ es 1, entonces existen $a, b \in \Lambda$ tal que $\gcd(a, b) = 1$, el número de agujeros de $\langle a, b\rangle$ es finito por la Proposición 3.37 y Λ contiene a $\langle a, b\rangle$, por lo que el numero de agujeros de Λ es finito. □

Lema 3.39 Sea Λ un semigrupo con un número finito de agujeros, $s \in \Lambda$. Entonces el número de elementos de $\Lambda \setminus (s + \Lambda)$ es igual a s.

Prueba: Sea c el conductor de Λ y $T = \{t \in \mathbb{N}_0 | t \geq s + c\}$ entonces T está contenido en Λ y en $s + \Lambda$, ahora, sea $U = \{u \in \Lambda | u < s + c\}$, entonces el número de elementos de U es igual a $s + c - g$ y Λ es la unión de los conjuntos disjuntos T y U. Sea $V = \{v \in s + \Lambda | s \leq v < s + c\}$ por lo que el número de elementos de V es igual a $c - g$ y $s + \Lambda$ es la unión de los conjuntos disjuntos V y T. Además $V \subseteq U$, por lo que $s \in \Lambda$ y Λ es un semigrupo. Entonces

$$\#(\Lambda \setminus (s + \Lambda)) = \#U - \#V = (s + c - g) - (c - g) = s.$$

□

Lema 3.40 Sea f un elemento distinto de cero de un $\mathbb{F}_q$-álgebra R, con una función peso ρ, entonces

$$\dim(R/f) = \rho(f).$$

Prueba: Sea Λ el semigrupo de la función peso ρ, y $s = \rho(f)$. Sea $(\rho_i | i \in \mathbb{N})$ la secuencia de los elementos de Λ en orden creciente. La imagen por ρ del conjunto de elementos distintos de cero del ideal (f) es igual a $s + \Lambda$. Entonces para cada $\rho_i \in \Lambda$ existe un $f_i \in R$ tal que $\rho(f) = \rho_i$, si además $\rho_i \in (s + \Lambda)$ entonces podemos seleccionar $f_i \in (f)$. Los conjuntos $\{f_i | i \in \mathbb{N}\}$ y $\{f_i | i \in \mathbb{N}, \rho_i \in (s + \Lambda)\}$ son bases del álgebra R y del ideal (f) respectivamente, por un argumento similar utilizado en la Proposición 3.13. Por ello, las clases $f_i \pmod f$, con $i \in \mathbb{N}$ y $\rho_i \in \Lambda \setminus (s + \lambda)$, forman una base de $R/(f)$, entonces la dimensión de $R/(f)$ es igual al número de elementos de $\Lambda/(s + \lambda)$ el cual es $\rho(f)$ por el Lema 3.39. □

Lema 3.41 Sea R un álgebra afín con función peso ρ y un mapeo de evaluación $ev_{\mathcal{P}}$, además f un elemento distinto de cero de R, entonces el número de ceros de f es a lo más $\rho(f)$.

Prueba: Sea $\mathcal{Q}$ el conjunto de ceros de f y sea $t = |\mathcal{Q}|$, el mapeo $ev_{\mathcal{Q}} : R \to \mathbb{F}_q^t$ es lineal y sobreyectivo por el Lema 3.18, además $g(Q) = 0, \forall Q \in \mathcal{Q}$ y $g \in (f)$. Esto induce un mapeo bien definido $ev_{\mathcal{Q}} : R \setminus (f) \to \mathbb{F}_q^t$ el cual es lineal y sobreyectivo, entonces el número de ceros de f es a lo más la dimensión de $R/(f)$ el cual es igual a $\rho(f)$ por el Lema 3.40. □

Supongamos ahora que se tiene un función peso ρ en $R = \mathbb{F}_q[X_1, X_2, \ldots, X_m]/I$ y sea $(\rho_i | i \in \mathbb{N})$ la enumeración de los elementos del semigrupo de ρ en forma creciente. Se considera el conjunto $\mathcal{P}$ que consiste en n puntos distintos de $\mathbb{F}_q^m$ en el conjunto de ceros de I, sea $ev_{\mathcal{P}} : R \to \mathbb{F}_q^n$ el correspondiente mapeo de evaluación. Tomamos entonces el siguiente código de evaluación: $E_l = \{ev_{\mathcal{P}}(f) | f \in R, \rho(f) \leq \rho_l\}$.

Teorema 3.42 La distancia mínima de E_l es al menos $n - \rho_l$. Si $\rho_l < n$ entonces $\dim(E_l) = l$.

Prueba: Sea c un elemento distinto de cero de E_l, entonces existe un elemento distinto de cero $f \in R$ tal que $\rho(f) \leq \rho_l$ y $c = ev_{\mathcal{P}}(f)$. Por ello $c_i = f(P_i)$, para todo i. Dado que el número de ceros de f es a lo más ρ_l por el Lema 3.41, entonces $\mathrm{wt}(c) \leq n - \rho_l$.

Supongamos ahora que $\rho_l < n$. E_l es la imagen sobre el mapeo de evaluación del espacio vectorial L_l de dimensión l. Si $f \in L_l$ y $ev_{\mathcal{P}}(f) = 0$, entonces f tiene al menos n ceros, por lo que $f = 0$ por el Lema 3.41, ya que se tiene que $\rho_l < n$. Entonces el mapeo $ev_{\mathcal{P}} : L_l \to E_l$ es un isomorfismo lineal, por tanto $\dim(E_l) = l$. □

Colorario 3.43 Sea ρ una función peso con g agujeros, si $\rho_k < n$ entonces E_k es un código $[n, k, d]$ tal que $k + d > n + 1 - g$.

Prueba: Esto se sigue del teorema anterior y del hecho que $\rho_k \leq k + g - 1$ como se muestra en el Lema 3.33. □

Las cotas de orden se estudian cuando ρ es una función peso en términos de su semigrupo asociado, cuando es generado por dos generadores, y, más general, para semigrupos llamados telescópicos.

3.4.2. Semigrupos y la distancia mínima dual

Sea ρ una función de peso en la $\mathbb{F}$-álgebra R, se asume que el máximo común divisor de los elementos de la función de peso $\rho(f), f \in R, f \neq 0$, es 1. Entonces el número de agujeros g correspondientes al semigrupo Λ es finito. Sea $(\rho_i | i \in \mathbb{N})$ la secuencia de los no agujeros de la función peso ρ tal que $\rho_i < \rho_{i+1}$, para todo i. Además el número de agujeros menores a ρ_l es denotado por $g(l)$ y el conductor de Λ se denota por c.

Para una función de peso ρ la función $l(i, j)$ es determinada por

$$\rho_{l(i,j)} = \rho_i + \rho_j.$$

Ahora N_l se define como

$$N_l = \{(i, j) \in \mathbb{N}^2 | \rho_i + \rho_j = \rho_{l+1}\} \text{ y}$$

v_l denota el número de elementos de N_l.
Recordemos que la cota de orden se define como

$$d(l) = \text{mín}\{v_m | m \geq l\}.$$

Definición 3.44 La cota de Goppa en la distancia mínina de C_l se denota por $d_G(l)$ y se define como $d_G(l) = l + 1 - g$.

Teorema 3.45 Sea $D(l) = \{(x, y) | x, y \text{ son agujeros con } x + y = \rho_{l+1}\}$, entonces

$$v_l = l + 1 - g(l + 1) + \#D(l),$$

donde $g(l + 1) = g$. Si $l \geq c - g$ y $\#D(l) = 0$. Si $l > 2c - g - 2$. Además $d(l) \geq d_G(l) = l + 1 - g$. Por lo que la igualad se cumple cuando $l > 2c - g - 2$.

Prueba: Para algún entero l dado, se define el conjunto $A(l)$ como el conjunto de pares de enteros no negativos (x, y) tales que $x + y = \rho_{l+1}$. Sea $B(l)$ el conjunto de pares (x, y) que pertenecen a $A(l)$ tales que x es un agujero y sea $C(l)$ el conjunto de pares $(x, y) \in A(l)$ tales que y es un agujero. Se tiene que $A(l) = N_l \cup B(l) \cup C(l)$ y $D(l) = B(l) \cap C(l)$ y N_l es disjunto de $B(l) \cup C(l)$ entonces

$$v_l = \#A(l) - \#B(l) - \#C(l) + \#D(l).$$

El número de elementos de $A(l)$ es $\rho_{l+1} + 1$. Sea $x \in \mathbb{N}_0$, entonces x es el agujero más pequeño que ρ_{l+1} si y solo si existe un único y tal que $(x, y) \in B(l)$. Entonces $\#B(l) = g(l + 1)$ y de forma similar $\#C(l) = g(l + 1)$. La igualdad para v_l se sigue de que $g(l + 1) = \rho_{l+1} - l$ por el Lema 3.33.

Si $l \geq c - g$, entonces $g(l + 1) = g$ por Lema 3.33(3).

Si $l > 2c-g-2$ y $g = 0$ entonces $v_l = l+1$ para todo $l \in \mathbb{N}$. Si $g > 0$ y $c \leq 2$ entonces $2c-g-2 \geq c-g$, por lo que $\rho_{l+1} = l + g > 2c - 2$. Sean x, y agujeros, entonces $x, y \leq c - 1$, si además $x + y = \rho_{l+1}$, entonces $\rho_{l+1} \leq 2c-2$, por tanto tal par (x, y) no existe. Como consecuencia $D(l)$ es vacío si $l > 2c-g-2$.□

Proposición 3.46 Sea ρ una función peso y g el número de agujeros del correspondiente semigrupo, entonces $r_\rho(d) \leq r(d) \leq d - 1 + g$.

Prueba: La desigualdad $r_\varphi(d) \leq r(d)$ se sigue de la inclusión de $R_\varphi(d) \subseteq R(d)$.

Si $l \in \mathbb{N}_0$ y $l \geq d - 1 + g$ entonces $v_l \geq l + 1 - g$ por el Teorema 3.45, luego se tiene que $l + 1 \notin R(d)$. En consecuencia $R(d) \subseteq \{1, 2, \ldots, d - 1 + g\}$ y $r(d) \leq d - 1 + g$. □

Lema 3.47 Sean a, b dos enteros positivos relativamente primos tales que $a > b$. Sea Λ el semigrupo generado por a y b. Entonces escribimos $\rho_{l+1} = bx + ay$ para algunos enteros no negativos x, y. Si $\rho_{l+1} < (b-1)a$, entonces $v_l = (x+1)(y+1)$ y existe al menos un agujero en el intervalo $[\rho_{l+1} - v_l, \rho_{l+1}]$.

Prueba: Sea $m = \rho_{l+1}$ y $v = v_l$, entonces v es el número de pares $(m_1, m_2) \in \Lambda^2$ tales que $m_1+m_2 = m$, se usara que si $m' \in \Lambda$ y $m' < (b-1)a$ entonces $y < b$, y existen únicos enteros no negativos x, y determinados tales que $m = bx + ay$, puesto que $\gcd(a, b) = 1$.

Consideremos ahora varios casos.

1. Sea $(i, j) \in \mathbb{N}_0^2$ tales que $0 \leq i \leq x$ y $0 \leq j \leq y$, se define $m_1(i, j) = bi + aj$ y $m_2(i, j) = (x - i)b + (y - j)a$, entonces $m_1(i, j), m_2(i, j) \in \Lambda$ y $m_1(i, j) + m_2(i, j) = m$. Los $m_1(i, j)$ son mutuamente distintos y por eso $v \leq (x + 1)(y + 1)$.

   Si (m_1, m_2) es un par tal que $m = m_1+m_2$ y $m_1, m_2 \in \Lambda$ entonces $m_1 = bx_1+ay_1$ y $m_2 = bx_2+ay_2$ para algunos enteros no negativos x_1, y_1, x_2 y y_2, entonces $bx + ay = (x_1 + x_2)b + (y_1 + y_2)a$. Luego $x_1 + x_2 = x$ y $y_1 + y_2 = y$, esto es que $m_t = m_t(x_t, y_t)$ para $t = 1, 2$. Deducimos entonces que $v = (x + 1)(y + 1)$.

2. Sea $m - i$ un elemento del intervalo $[m - v, m]$ y

   $$m - i = bx_i + ay_i, 0 \leq y_1 \leq b \text{ para } i = 0, 1, \ldots, v.$$

   Si se demuestra que uno de estos x_i es negativo, entonces existe al menos un agujero en $[m-v, m]$. Para verlo considérese dos casos:

Caso I $v < b$, aquí los y_i, con $i = 0, 1, \ldots, v$ son $v+1$, enteros no negativos distintos, por lo que existe al menos un $y_i \geq v = (x + 1)(y + 1)$. Para el correspondiente x_i se tiene que

$$x_i b = m - iay_i \leq bx + ay - i - (x + 1)(y + 1)a \leq x(b - a) < 0,$$

puesto que $b < a$.

Caso II $v \geq b$, entonces $m - i$ toma todos los valores posibles módulo b, además $m - i \equiv ay_i \pmod{b}$ y $\gcd(a, b) = 1$. Por ello, y_i toma todos los valores posibles módulo b. Por lo que encontramos $y_i = b - 1$ para algún $i = 0, 1, \ldots, v$. Para el correspondiente x_i se tiene que

$$bx_i = m - i - ay_i \leq m - (b-1)a < 0,$$

siendo que se asumió que $m < (b-1)a$.
En ambos casos se muestra que uno de los x_i es negativo. □

Proposición 3.48 Se considera el semigrupo Λ de funciones peso generado por los elementos a y b tales que $b < a$ y $\gcd(a, b) = 1$. Sea (ρ_i) una enumeración en forma creciente del semigrupo, entonces, con la notación anterior, se cumple que:

$$d(l) = j + 1 \text{ si } l < g \text{ y } (j-1)a < \rho_{l+1} \leq ja.$$

Prueba: El semigrupo es simétrico, entonces $c = 2g$ y $c = (a-1)(b-1)$ por la Proposición 3.37. Si $l < g$, entonces $l < c - g$, por el Lema 3.33 se tiene que $\rho_{l+1} < c - 1$, esto es que $\rho_l < (b-1)a$. Escribiremos $\rho_{l+1} = bx + ay$ para algunos enteros no negativos x, y, por lo que $v_l = (x+1)(y+1)$ por el Lema 3.47. Si además $\rho_{l+1} = aj$, entonces $x = 0$ y $y = j$, luego $v_l = j + 1$. Supongamos que $(j-1)a < \rho_{l+1} = bx + ay < aj$, entonces $0 \leq y \leq j - 1$. Entonces $v_l = (x+1)(y+1)$ es estrictamente mayor que

$$\left(\frac{(j-1-y)a}{b} + 1\right)(y+1) \geq (j-1-y) + (y+1) = j,$$

por lo que $d(l) = \min\{v_m | m \geq l\} = j + 1$. □

Teorema 3.49 Sea Λ como en la Proposición 3.48. Si $l \geq g$, entonces

$$d(l) = \min\{\rho_t | \rho_t \geq l + 1 - g\}.$$

Prueba: Λ es simétrico por la Proposición 3.37, entonces $c = 2g$. Si $l \geq g = c - g$, entonces $\rho_{l+1} = l + g$ por el Lema 3.33.

- Si $l < 3g - 2 = 2c - g - 2$, entonces $d(l) = l + 1 - g$ por el Teorema 3.45. Además $l - 2g + 2 > g = c - g$. Entonces $\rho_{l-2g+2} = l + 1 - g$, por lo que $d(l) = \rho_{l-2g+2}$.
- Si $g \leq l \leq 3g - 2$, entonces $\rho_{l+1} = l + g$. Por tanto $2g \leq \rho_{l+1} \leq 4g - 2$, podemos escribir $\rho_{l+1} = 2g - 1 + k, 1 \leq k \leq 2g - 1$, esto es que $k = l + 1 - g$. El número v_l es igual a $l + 1 - g + \#D(l)$, por el Teorema 3.45, para la estimación del número de elementos de $D(l)$ se consideran dos casos:

Caso II k no es un agujero. Sea $(x, y) \in D(l)$, entonces x, y son agujeros y $\rho_{l+1} = x + y$, luego $(2g - 1 - x) + k = y$, el semigrupo es simétrico, entonces $2g - 1 - x$ no es un agujero. Ya que la suma de dos no agujeros es de cuenta un no agujero, entonces y no puede ser un agujero y $D(l)$ es vacío. En consecuencia $v_l = l + 1 - g = k$ es un no agujero ρ_t para algún t.

Caso II k es un agujero, existe un t tal que $\rho_{t-1} < k < \rho_t$ y un $L \geq l$ tal que $\rho_{L+1} = 2g - 1 + \rho_t$. Teniendo en cuenta el argumento en el caso anterior, tenemos que $v_l = L + 1 - g = \rho_t$. Se debe mostrar que $v_l \geq \rho_t$.
La función $\#D(l)$ es definida como una condición en los agujeros, pero para semigrupos simétricos tal condición puede ser trasladada como una condición en los no agujeros. Definimos $x' = 2g - 1 - x$ si $x \in \mathbb{N}_0$, $0 \leq x \leq 2g - 1$. Entonces:

$$x, y \in (\mathbb{N}_0 \setminus \Lambda), x + y = \rho_{l+1} \text{ si y solo si } x', y' \in \Lambda, x' + y' = 2g - 1 - k.$$

En consecuencia $2g - 1 - k$ es un no agujero ρ_{u+1} y el número de elementos de $D(l)$ es igual a v_u. Por lo tanto, existe un agujero en el intervalo $[\rho_{u+1} - v_u, \rho_{u+1}]$. Por el Lema 3.47 se tiene que $\rho_{t-1} < k < \rho_t$, entonces los números $k, k + 1, \rho_t - 2, \rho_t - 1$ son todos agujeros. Entonces

$$2g - \rho_t, 2g + 1 - \rho_t, \ldots, 2g - 2 - k, 2g - 1 - k = \rho_{u+1}$$

son todos no agujeros. Por lo que $\rho_{u+1} - v_u < 2g - \rho_t$, pero $\rho_{u+1} = 2g - 1 - k$. Por ello $\rho_t - k \leq v_u = \#D(l)$, esto es que $v_l = l + 1 - g + \#D(l) \geq \rho_t$ y $k = l + 1 - g$, esto implica que $d(l) = \rho_t$ y k es el no agujero más pequeño el cual es menor que $l + 1 - g$. □

3.4.3. Semigrupos telescópicos

Definición 3.50 Sea $(a_1, a_2, \ldots, a_k)$ una secuencia de números positivos con máximo común divisor 1. Se define

$$d_i = \gcd(a_1, \ldots, a_i) \text{ y } A_i = \left\{ \frac{a_1}{d_1}, \ldots, \frac{a_1}{d_i} \right\},$$

para $i = 1, \ldots, k$. Sea $d_0 = 0$ y Λ_i el semigrupo generado por A_i. Si $\frac{a_i}{d_i} \in \Lambda_{i-1}$ para $i = 2, \ldots, k$. Entonces la secuencia $(a_1, a_2, \ldots, a_k)$ es llamada telescópica.

Definición 3.51 Un semigrupo telescópico es aquel semigrupo generado por una sucesión telescópica.

- Si $(a_1, a_2, \ldots, a_k)$ es una secuencia telescópica, entonces $\gcd\left(\frac{a_1}{d_i}, \ldots, \frac{a_i}{d_i}\right) = 1$ y la secuencia $\left(\frac{a_1}{d_i} \ldots, \frac{a_i}{d_i}\right)$ es telescópica para $i = 2, \ldots, k$.
- Si $d_i = 1$ para una secuencia telescópica $(a_1, a_2, \ldots, a_k)$, entonces $(a_1, a_2, \ldots, a_i)$ es también telescópica y generan el mismo semigrupo.

Lema 3.52 Si $(a_1, a_2, \ldots, a_k)$ es una sucesión telescópica y además $m \in \Lambda_k$. Entonces existen enteros no negativos determinados unicamente $x_1, x_2, \ldots, x_k$ tales que $0 \leq x_i < \frac{d_{i-1}}{d_i}$ para $i = 2, \ldots, k$ y

$$m = \sum_{i=1}^{k} x_i a_i.$$

Esta representación es llamada la representación normal de m por $(a_1, a_2, \ldots, a_k)$.

Prueba: Por inducción sobre el número k de entradas en la secuencia. Para $k = 1$ no hay nada que probar. Para $k = 2$ el lema dice: si $\gcd(a_1 a_2) = 1$, entonces cada $m \in \Lambda_2$ puede ser escrita unicamente como $m = a_2 x_2 + a_1 y_1, 0 \leq x_2 \leq a_1$. Este hecho es usado en la prueba de la Proposición 3.37. Ahora el lema se asume cierto para todas las secuencias telescópicas con $k-1$ entradas y $m \in \Lambda_k$. Existe $x_k \in \mathbb{N}_0$ y $u \in \Lambda_{k-1}$ tal que $m = a_k x_k + d_{k-1} u$. Entonces $\Lambda_k = \langle a_k \rangle + d_{k-1}\Lambda_{k-1}$. Escribiendo $x_k = d_{k-1} w + v, 0 \leq v \leq d_{k-1}$ se obtiene que $m = a_k v + d_{k-1}(u + a_k w)$. Luego $a_k \in \Lambda_{k-1}$ y en consecuencia $u + a_k w \in \Lambda_{k-1}$. Tomando en cuenta que $\left(\frac{a_1}{d_{k-1}}, \frac{a_{k-1}}{d_{k-1}}, \ldots, \frac{a_1}{d_{k-1}}\right)$ es telescópica. Sea $d'_i = \gcd\left(\frac{a_1}{d_{k-1}}, \frac{a_2}{d_{k-1}}, \ldots, \frac{a_{k-1}}{d_{k-1}}\right)$ para $i = 1, 2, \ldots, k-1$. Por tanto existen $0 \leq x_i < \frac{d'_{i-1}}{d'_i}$ para $i = 2, 3, \ldots, k-1$, tales que

$$u + a_k w = \sum_{i=1}^{k-1} x_i \frac{a_i}{d_{k-1}}$$

es una representación normal de $\left(\frac{a_1}{d_{k-1}}, \frac{a_2}{d_{k-1}}, \ldots, \frac{a_{k-1}}{d_{k-1}}\right)$. Entonces $m = a_k v + \sum_l^{k-1} a_i x_i$ es una representación normal para $(a_1, a_2, \ldots, a_k)$. Por lo que $\frac{d'_{i-1}}{d'_i} = \frac{d_{i-1}}{di}$.

Para probar la unicidad asumamos que m tiene dos representaciones normales y sean estas $\sum_{i=1}^{k} a_i x_i = m = \sum_{i=1}^{k} a_i y_i$, donde $0 \leq x_i, y_i < \frac{d_{i-1}}{d_i}$ para $i = 2, 3, \ldots, k$. Sea l el mayor índice para el cual $x_i \neq y_i$. Entonces $\sum_{i=1}^{l} a_i x_i = \sum_{i=1}^{l} a_i y_i$ y $(x_l - y_l)a_l = \sum_{i=1}^{l-1}(y_i - x_i)a_i$. Por tanto, el lado derecho es un múltiplo de d_{l-1} y el $\gcd(\frac{a_l}{d_l}, \frac{d_{l-1}}{dl}) = 1$. Finalmente, se tiene que $x_l - y_l$ es un múltiplo no nulo de $\frac{d_{l-1}}{d_l}$ lo cual es una contradicción. □

Proposición 3.53 Sea Λ_k el semigrupo generado por la sucesión telescópica $(a_1, a_2, \ldots, a_k)$. Entonces

$$c(\Lambda_k) - 1 = d_{k-1}(c(\Lambda_{k-1}) - 1) + (d_{k-1} - 1)a_k = \sum_{i=1}^{k} \left(\frac{d_{i-1}}{d_i - 1}\right) a_i,$$

$$g(\Lambda_k) = d_{k-1}(\Lambda_{k-1}) + \frac{(d_{k-1}-1)(a_k-1)}{2} = \frac{c(\Lambda_k)}{2}.$$

Entonces los semigrupos telescópicos son simétricos (se ha usado $d_0 = 0$).

Prueba: Si $k = 1$, entonces $\Lambda_1 = \mathbb{N}_0$. Por lo que el conductor es cero y el número de agujeros es cero. Esto es para estar en concordancia con las formulas. Para $k = 2$ obtenemos la Proposición 3.37.
Asumamos $k > 1$. Como $\gcd(a_k, d_{k-1}) = 1$. Cada entero $m \in \mathbb{N}_0$ puede ser representado unicamente como $m = a_k v + d_{k-1}w$, $0 \leq v < d_{k-1}$. Notemos que w puede ser negativo. Entonces por el Lema 3.52 los agujeros de Λ_k son exactamente los números m, donde el correspondiente w es un agujero Λ_{k-1} o w es negativo. Como la primer ecuación involucra al mayor agujero $c(\Lambda_k) - 1$ en término de los conductores $c(\Lambda_k)$ se obtiene el resultado.

Para la segunda igualdad se procede por inducción. Para cada valor de $0 \leq v < d_{k-1}$ hay $g(\Lambda_{k-1})$ agujeros de Λ que son estos los de Λ_{k-1}. Además, obtenemos los agujeros de la forma $m = a_k v + d_{k-1}w$, donde $w < 0$, pero estos agujeros son exactamente los agujeros del semigrupo $\langle a_k, d_{k-1}\rangle$. Este número es

$$\frac{(d_{k-1}-1)(a_k-1)}{2},$$

por la Proposición 3.37. Entonces el número total de agujeros es igual a

$$d_{k-1}g(\Lambda_{k-1}) + \frac{(d_{k-1}-1)(a_k-1)}{2}.$$

El resto del resultado sobre la simetría es obtenido por inducción.

3.5. Decodificación de códigos geométrico algebraicos

Estudiamos brevemente dos algoritmos de decodificación para el código de evaluación C_l. Uno de ellos es el algoritmo básico. Este algoritmo corrige hasta $\left\lfloor \frac{d_G(l) - 1 - g}{2} \right\rfloor$ errores con ρ como una función peso con g agujeros. El segundo algoritmo es el algoritmo denominado por mayoría de votos con síndrome desconocido. Este algoritmo permite decodificar hasta la mitad de la cota de orden si ρ es una función de orden arbitraria.

3.5.1. El problema

Sea C un código lineal en $\mathbb{F}_q^n$ con distancia mínima d. Si c es una palabra transmitida y $c + e$ es la palabra recibida, donde e se denomina el vector error. Además, $\{i | e_i \neq 0\}$ es el conjunto de posiciones de errores. Los elementos e_i son llamados los *valores de error* y $\mathrm{wt}(e)$ es el número de errores en la palabra recibida. Si y es la palabra recibida donde la distancia de y al código C es t'. Entonces existe una palabra de código c' y un vector de errores e' tal que $y = c' + e'$ y $\mathrm{wt}(e') = t'$. Si el número de errores es a lo mas $\frac{d-1}{2}$, entonces $c{=}c'$ y $e{=}e'$. Esto significa que la palabra del código más cercana a y es única cuando y tiene distancia a lo más $\frac{(d-1)}{2}$ al código C.

Definición 3.54 Un decodificador es un mapeo $\mathcal{D} : \mathbb{F}_q^n \to C*$, donde $C* = C \cup \{?\}$
$\mathcal{D}$ es un decodificador del código C si $\mathcal{D}(c) = c, \forall c \in C$.

Una posible imagen es el símbolo *?*, este es el resultado cuando se falla en encontrar una palabra del código.

Definición 3.55 Un decodificador de máxima probabilidad para el código C es un decodificador $\mathcal{D}$ tal que $\mathcal{D}(y)$ es la palabra más cercana a C para todo y.

Definición 3.56 Un decodificador de distancia acotada es un decodificador $\mathcal{D}$ para el código C que corrige t errores. Si $\mathcal{D}(y)$ es la palabra más cercana para todo $y \in \mathbb{F}_q^n$ tal que $d(y, C) \leq t$.

Definición 3.57 Un decodificador de distancia mínima es un decodificador $\mathcal{D}$ para el código C de distancia mínima d, que decodifica hasta la mitad de la distancia mínima si $\mathcal{D}(y)$ es la palabra más cercana para todo $y \in \mathbb{F}_q^n$ tal que $d(y, C) = \frac{d-1}{2}$.

Los errores pueden corregirse al solucionar un sistema de ecuaciones lineales que involucran síndromes si se tiene la información completa acerca de la posición de los errores. Esto se cumple si se tienen solo borrones en la información.

Proposición 3.58 Sea C un código lineal en $\mathbb{F}_q^n$ con matriz de verificación de paridad H. Si se tiene una palabra recibida y con vector de errores e y conocemos un conjunto J con a lo mas $d(C) - 1$ elementos que contienen al conjunto de posiciones de error. Entonces el vector de errores e es la única solución de la siguientes ecuaciones lineales:

$$xH^T = yH^T \text{ y } x_j = 0, \text{ para todo } j \notin J.$$

Prueba: De forma inmediata se tiene que el vector e es solución del sistema. Por otra parte, si x es otra solución, entonces $(x - e)H^T = 0$. Por lo que $x - e$ es un elemento de C y además está soportado en J. En consecuencia, estos pesos son a los mas $d(C) - 1$. Entonces el peso es a lo más $d(C) - 1$. En consecuencia debe ser cero. Por lo que $x = e$. □

Por lo anterior, el problema de decodificación de una palabra recibida y con errores se puede reducir al problema de encontrar las posiciones de estos errores. Si se desea decodificar todas las palabras recibidas con t errores, entonces existen $\binom{n}{t}$ posibles t-conjuntos de posiciones de errores a considerar. Este número crece exponencialmente en n cuando $\frac{t}{n}$ tiende a un número real distinto a cero.

Definición 3.59 Un sistema recubridor es una colección $\mathcal{J}$ de subconjuntos J de $\{1, 2, \ldots, n\}$, tal que para todo $J \in \mathcal{J}$ este tiene $d-1$ elementos y cada subconjunto de $\{1, 2, \ldots, n\}$ de tamaño t esta contenido en al menos un $J \in \mathcal{J}$.

De la proposición anterior, para decodificar todas las palabras recibidas es suficiente encontrar un $(n, d-1, t)$ sistema recubridor. El tamaño de tal sistema recubridor es pequeño en comparación al número de todos los posibles t-conjuntos, pero este número es al menos $\binom{n}{t}/\binom{d-1}{t}$.

3.5.2. El algoritmo básico

Sea ρ una función de orden en una $\mathbb{F}_q$-álgebra afín R y $\varphi : R \to \mathbb{F}_q^n$ un morfismo sobreyectivo. Además, $\{f_i | i \in \mathbb{N}\}$ es una base de R sobre $\mathbb{F}_q$ tal que $\rho(f_i) < \rho_{f_{i+1}}, \forall i \in \mathbb{N}$. Sea L_i el espacio vectorial generado por $\{f_1, f_2, \ldots, f_l\}$. Entonces se define $l(i, j)$ como el entero positivo más pequeño l tal que $f_i f_j \in L_l$. Si $h_i = \varphi(f_i)$ tenemos que $C_l = \{c \in \mathbb{F}_q^n | c.h_i = 0, \forall i \leq l\}$. Por ello $h_i * h_j$ verifica la paridad de C_l si $l(i, j) \leq l$.

Los síndromes son definidos como $s_i(y) = y.h_i$ y $s_{ij}(y) = h.(h_i * h_j)$. Además, denotamos como $S(i, j)$ a la submatriz de tamaño $i \times j$ obtenida de la matriz de síndromes

$$S(y) = S(i, j) = (s_{i'j'}(y) | 1 \leq i' \leq i, 1 \leq j' \leq j).$$

Si y es una palabra recibida y $y = c + e$ con $c \in C_l$. Entonces $s_{i,j}(y) = s_{i,j}(e)$ para todo i, j tales que $l(i, j) \leq l$.

Definición 3.60 Si $y \in \mathbb{F}_q^n$ con $l(i, j) \leq l$. Por ello se define el espacio

$$K_{ij}(y) = \{f \in L_j | y.\varphi(fg) = 0, \forall g \in L_i\}.$$

En consecuencia, K_{ij} es un subespacio de L_l. K_{ij} es el núcleo del mapeo lineal $L_j \to L_i$ y matriz $S(i, j)$ con respecto a las bases $\{f_1, f_2, \ldots, f_j\}$ y $\{f_1, f_2, \ldots, f_i\}$ de los espacios vectoriales L_j y L_i, respectivamente. Por lo que $K_{ij}(y) = K_{ij}(e)$.

Definición 3.61 Si J es un subconjunto de $\{1, 2, \ldots, n\}$. Entonces se define el subespacio

$$L_j(J) = \{f \in L_j | \varphi(f)_k = 0, \forall k \in J\},$$

tal que $\varphi(f)_k$ es la k-ésima coordenada de $\varphi(f)$.

Lema 3.62 Si $I = \text{sop}(e) = \{k \in \{1, 2, \ldots, n\} | e_k \neq 0\}$. Entonces $L_j(I) \subseteq K_{ij}(y)$ y si además $d(C_i) > \text{wt}(e)$. En consecuencia $L_j(I) = K_{ij}(y)$.

Prueba: Sea $f \in L_j(I)$, entonces $\varphi(f)_k = 0$ para todo k tal que $e_k \neq 0$. Por tanto

$$e.(\varphi(f) * \varphi(g)) = \sum_{e_k \neq 0} e_k(\varphi(f) * \varphi(g))_k = 0,$$

para todo $g \in L_i$. Esto es que $f \in K_{ij}(e) = K_{ij}(y)$.
Si $d(C_i) > \text{wt}(e)$ y $f \in K_{ij}(y)$ con $a = \varphi(f)$. Entonces $f \in K_{ij}(e)$. Por ello

$$(e * a).\varphi(g) = e.(\varphi(f) * \varphi(g)) = 0,$$

para todo $g \in L_i$, dando $e * a \in C_i$. Con esto $\text{wt}(e * a) \leq \text{wt}(e) < d(C_i)$ y $e * a = 0$. Esto es que $e_k\varphi(f)_k = 0, \forall k \in \{1, 2, \ldots, n\}$. En consecuencia $\varphi(f)_k = 0, \forall k \in I = \text{sop}(e)$ y finalmente $f \in L_j(I)$. □

Sea I el conjunto de las posiciones de error del $\text{sop}(e)$. El conjunto de las coordenadas cero de $\varphi(f)$, donde $f \in L_j(I)$ contiene el conjunto de posiciones de error. Por esta razón los elementos de $L_j(I)$ son llamados **funciones localizadores de error**. Pero el espacio $L_j(I)$ no es conocido. El espacio $K_{ij}(y)$ puede ser calculado después de recibida la palabra y. La igualdad $L_j(I) = K_{ij}(y)$ implica que todos los elementos de $K_{ij}(y)$ son funciones localizadoras de errores.

Más generalmente, cada elemento f de R satisface que $\varphi(f)_k = 0$, para todo $k \in \text{sop}(e)$ es llamado un **localizador de error** que constituyen un ideal L de R. Si $\text{wt}(e) = t$ entonces la dimensión de R/L como un espacio vectorial $\mathbb{F}_q$, es t .

Si $l(i, j) \leq l$. El algoritmo básico $\mathcal{A}(i, j)$ para el código $C = C_l$ calcula el núcleo $K_{ij}(y)$ para cada palabra recibida y. Si este núcleo es distinto de cero, este toma un elemento distinto de cero f y determina el conjunto J de posiciones de cero en f. Si $d(C_i) > \text{wt}(e)$, donde e es el vector de errores, entonces J contiene el soporte de e y por el Lema 3.62. Si el conjunto J no es muy grande, entonces se aplica la Proposición 3.58 para obtener los valores esperados.

Con esto se tiene un algoritmo básico para cada par (i, j) tales que $l(i, j) \leq l$. Si j es pequeño con respecto al número de errores, entonces $K_{ij}(y) = 0$. Si j es grande, entonces i puede ser pequeño, que resulta en un código grande C_i y esto dificulta reunir los requerimientos que $d(C_i) > \text{wt}(e)$.

Proposición 3.63 Sea ρ una función peso con g agujeros. En consecuencia, el algoritmo básico corrige $\lfloor \frac{d_G(l) - 1 - g}{2} \rfloor$ errores para el código C_l con complejidad $O(n^3)$.

Prueba: Se asume que $t = \lfloor \frac{d_G(l) - 1 - g}{2} \rfloor \geq 1$, entonces $l \geq 2g + 2$ y $\rho_l = l + g - 1$. Además, por simplicidad tomemos l como un número par. La distancia mínima de Goppa $d_G(l) = l + 1 - g$. Entonces $t = l/2 - g$. Además, sea $j = t + 1$ y sea $i = \frac{l}{2}$. Entonces $\rho_j \leq \frac{l}{2}$ y $\rho_i = \frac{l}{2} + g - 1$. por lo que $\rho_i + \rho_j \leq l + g - 1 \leq \rho_l$. En consecuencia $l(i, j) \leq l$ y el algoritmo básico $\mathcal{A}(i, j)$ puede ser aplicado para decodificar C_l.

Si y es una a palabra recibida con a lo más t errores, entonces el vector de errores e con soporte I tiene tamaño a lo más t y $L_j(I)$ no es cero, además, I impone a lo más t condiciones lineales en L_j y la dimensión de L_j es $j = t + 1$.

siguiendo con la prueba, sea f un elemento f distinto de cero de $K_{ij}(y)$. El Teorema 3.45 implica que $d(C_i) \geq i + 1 - g$ la cual es estrictamente mayor que t. Entonces $K_{ij}(y) = L_j(I)$ por el Lema 3.62. Por lo que f es una función localizador de errores.

La función f tiene a lo más ρ_j ceros, por el Lema 3.41 ya que $\rho(f) \leq \rho_j$. Tomemos $J = \{k | f(P_k) = 0\}$. Entonces por el Lema 3.62 J contiene a I, el soporte de e. El número de elementos de J es a lo mas $\rho_j = \frac{l}{2} < l + 1 - g$, dado que $l > 2g$. Así $\#J < d(C_l)$ y la Proposición 3.58 da los valores de los errores.

La complejidad esta dada por la cantidad de operaciones simples que hay en resolver un sistema de ecuaciones lineales, esto es $O(n^3)$ [Bach and Shallit, 1996].

3.5.3. Votación por mayoría de síndromes desconocidos

Sea y una palabra recibida con vector de error e con respecto al código C_l. Si conocemos los síndromes $s_i = s_i(e)$ para todo $i \leq N$ con N definido como en la prueba del Lema 3.18, entonces podemos resolver el sistema de ecuaciones lineales $s_i(x) = s_i$, para todo i, el cual puede tener solución única $x = e$. Los síndromes $s_i(y)$ pueden ser calculados para todo i con $s_i(y) = s_i(e)$, para todo $i \leq l$. Entonces el síndrome $s_i(e)$ es llamado **conocido** con respecto a C_l si $i \leq l$ y **desconocido** si $i > l$. Esto muestra como los síndromes desconocidos s_{l+1} pueden ser obtenidos de los conocidos por **voto por mayoría**, si el número de errores es a lo más $\lfloor \frac{v_l - 1}{2} \rfloor$.

La matriz de síndromes $(s_{ij}(e)|1 \leq i, j \leq N)$ con respecto a un vector de errores e, que fue definido anteriormente como

$$s_{ij}(e) = e.\varphi(f_i f_j).$$

Si y es una palabra recibida con vector de errores e con respecto al código C_l y $l(i,j) \leq l$, entonces $f_i f_j \in L_l$, por lo que $s_{ij}(e) = s_{ij}(y)$. Así $s_{ij}(e)$ es una entrada conocida de la matriz de síndromes para todo i, j tales que $l(i,j) \leq l$. Para continuar, abreviamos $s_{ij}(e)$ y $s_l(e)$ por s_{ij} y s_l respectivamente.
El conjunto N_l fue definido anteriormente como

$$N_l = \{(i,j) \in \mathbb{N}^2 | l(i,j) = l+1\}.$$

Las entradas en la matriz de síndromes con índice $(i,j) \in N_l$ son los primeros síndromes desconocidos en encontrar con respecto al código C_l. Tan pronto se conozca algún s_{ij} con $(i,j) \in N_l$, se conocerán todos los otros $s_{i'j'}$ con $(i',j') \in N_l$, por lo que cada una de las funciones $f_i f_j, f_{i'} f_{j'}$ o f_{l+1} es un generador de un espacio vectorial L_{l+1} módulo L_l. Esto es que existe $\mu_{ij}, \mu_{ijk} \in \mathbb{F}_q$ tal que μ_{ij} es distinto de cero y

$$f_i f_j = \mu_{ij} f_{l+1} + \sum_{k=1}^{l} \mu_{ijk} f_k,$$

para todo i, j con $l(i,j) = l+1$. Por tanto

$$s_{ij} = \mu_{ij} s_{l+1} + \sum_{k=1}^{l} \mu_{ijk} s_k$$

y esta relación es la misma para todos los vectores de errores. Para continuar con el proceso considérese la matriz

$$S(i,j) = (s_{i'j'}(e)|1 \leq i' \leq i, 1 \leq j' \leq j,$$

como fue hecho para el algoritmo básico con los síndromes $s_{ij}(y)$ en lugar de $s_{ij}(e)$. Si $(i,j) = l+1$, entonces todas las entradas de la matriz a excepción de s_{ij} son conocidas. En consecuencia $l(i',j') \leq l$ si $i' \leq i, j' \leq j$ y $(i',j') \neq (ij)$.

$$\begin{pmatrix} s_{1,1} & \cdots & s_{1,j-1} & s_{1,j} \\ \vdots & & \vdots & \vdots \\ s_{i-1,1} & \cdots & s_{i-1,j-1} & s_{i-1,j} \\ s_{i,1} & \cdots & s_{i,j-1} & ? \end{pmatrix}$$

- Si $l(i,j) = l$, entonces $S(i,j)$ es una matriz del mapeo lineal de L_j a L_i el cual es usado para calcular el núcleo $K_{ij}(y)$ en el algoritmo básico $\mathcal{A}(i,j)$ para el código C_l.

- Si f es una función localizadora de errores distinta de cero en L_j y $f = \sum_{j'=1}^{j} \lambda_{j'} f_{j'}$, entonces las columnas de la matriz $S(i,j)$ son dependientes:

$$\sum_{j'=1}^{j} s_{i'j'} \lambda_{j'} = 0, \forall 1 \leq i' \leq i.$$

Definición 3.64 Si $(i,j) \in N_l$, esto es para decir que $l(i,j) = l+1$ y las 3 matrices $S(i-1,j-1), S(i-1,j)$ y $S(i,j-1)$ tienen igual rango, entonces (i,j) es llamado **un candidato** con respecto a C_l. Si (i,j) es un candidato, entonces existe un único valor s'_{ij} para asignarle a las entradas desconocidas s_{ij} tal que todas las matrices $S(i,j)$ y $S(i-1,j-1)$ tienen igual rango. Los elementos s'_{ij} es llamado **predicho** o **valor candidato** de los síndromes desconocidos s_{ij}. Un candidato es llamado **correcto** o **verdadero** cuando $s'_{ij} = s_{ij}$ e **incorrecto**(**falso**) en otro caso. Usando las identidades entre los síndromes, cada $(i,j) \in N_l$ da un valor predicho $s_{l+1}(i,j)$ de s_{l+1} por

$$s_{l+1}(i,j) = \frac{s'_{ij} - \sum_{k=1}^{l} \mu_{ijk} s_k}{\mu_{ij}}.$$

Se denota el número de candidatos verdaderos como T y el número de candidatos falsos como F. Una entrada (i,j) es llamada una **discrepancia** si las tres matrices $S(i-1,j-1), S(i-1,j)$ y $S(i,j-1)$ tienen igual rango y las matrices $S(i,j)$ y $S(i-1,j-1)$ no tienen igual rango.

- Las discrepancias son los *pivotes* si se aplica eliminación Gausiana (sin intercambiar filas o columnas) a la matriz de síndromes.
- El número total de discrepancias es igual al rango de la matriz de síndromes.
- Por el Lema 3.20, el rango de la matriz de síndromes es igual al peso de e.
- El número de discrepancias es igual al número de errores.

Sea y una palabra recibida con vector de errores e la cual tiene a lo mas $\dfrac{v_l - 1}{2}$ errores con respecto al código C_l. Se denota el número de discrepancias en la parte conocida de la matriz como K. Un candidato es incorrecto si y solo si es una discrepancia, entonces

$$K + F \leq \text{ número total de discrepancias } = \mathrm{wt}(e).$$

Si la entrada (i,j) es una discrepancia conocida, entonces todas las entradas (i',j') en la i-ésima fila con $j' > j$ y todas las entradas (i',j) en la j-ésima columna con $i' > i$ no son candidatos.
Si $(i,j) \in N_l$ no es un candidato, entones existe al menos una discrepancia conocida en la misma fila o columna. Por lo que el número de pares $(i,j) \in N_l$ que no son candidatos es a lo más $2K$.
El número de pares $(i,j) \in N_l$ los cuales son candidatos es igual a $T + F$. Por lo tanto

$$v_l = \#\text{ candidatos} + \#\text{ no candidatos} \leq (T+F) + 2K.$$

Asumiendo que el número de errores no son mas de $\frac{v_l - 1}{2}$. Entonces

$$\mathrm{wt}(e) \leq \frac{v_l - 1}{2}.$$

Combinando las anteriores inecuaciones, tenemos que

$$F < T.$$

No hay una forma directa de ver si un candidato es verdadero o es falso. Pero un valor predecido s'_{ij} del síndrome s_{ij} es asignado a cada candidato y este nos da una **predicción o voto** $s_{l+1}(i,j)$ para s_{l+1} por la Definición 3.64. Todos los T candidatos verdaderos producen lo mismo, valores correctos para s_{l+1}.

Proposición 3.65 Si el número de errores de una palabra recibida con respecto al código C_l es a lo mas $\frac{v_l - 1}{2}$, entonces la mayoría de los votos de los candidatos son para los valores correctos de s_{l+1}.

Por recursividad todos los síndromes desconocidos con respecto al código C_l pueden ser obtenidos si el número de errores es a lo mas $\lfloor \dfrac{d_\varphi(l) - 1}{2} \rfloor$, esto es que $v_m \geq d_\varphi(l)$ si $m \geq l$ y $C_{m+1} \neq C_m$. A partir de esto se obtiene el vector de errores.

Teorema 3.66 $\lfloor \frac{d_\varphi(l)-1}{2} \rfloor$ errores son corregidos para el código C_l usando la votación por mayoría para síndromes desconocidos con complejidad $O(n^3)$.

la complejidad es a lo más la complejidad de resolver un sistema de n ecuaciones lineales con n incógnitas, esto es , $O(n^3)$ [Bach and Shallit, 1996].

Concluimos este capítulo remarcando que los códigos lineales de evaluación son buenos códigos tal que si el objeto geométrico a tratar tiene género g mínimo, en especial genero 0 ($g = 0$), estos códigos son muy cercanos a la cota de Singleton que dice que $d \geq n - k + 1 - g$. Además, poseen algoritmos de descodificación óptimos, características de gran importancia en el siguiente capítulo.

Capítulo 4

Criptosistemas basados en códigos lineales correctores

Un mensaje es una sucesión ordenada de letras o símbolos de un cierto alfabeto. Si en un mensaje cifrado una o varias de estas letras son alteradas (sustituidas por otras letras del alfabeto), este error tenderá a confundir y desorientar al criptoanalista que trate de atacar tal mensaje cifrado. Esta observación sugiere utilizar la adición de errores como parte del proceso de cifrado. El mensaje alterado en principio tampoco podrá ser recuperado por el destinatario. La teoría introducida en el capítulo anterior acerca de códigos lineales de evaluación correctores nos hace afirmar que la eliminación de tales errores podría hacerse de manera eficiente si se utiliza en el criptosistema de McEliece con códigos lineales de evaluación.

En este capítulo se introduce el criptosistema de McEliece y sus variantes. Este criptosistema hace uso de códigos lineales y de algoritmos eficientes para la decodificación de mensajes. Nuestra propuesta es la utilización de códigos estudiados en el capítulo anterior ya que son lineales, tienen buena cota de corrección de errores y existen eficientes algoritmos de decodificación.

4.1. Criptosistema de McEliece

Sea C un $[n, k, d]$ código sobre el cuerpo finito $\mathbf{F}_q$. Supuesto emitido un mensaje c y recibido el mensaje alterado $y = c + e$, con $\mathrm{wt}(e) \leq t = \left\lfloor \frac{d-1}{2} \right\rfloor$, la eliminación del vector e, aunque teóricamente factible (c es el único elemento del código a distancia menor o igual que t de y), resulta computacionalmente imposible para parámetros suficientemente grandes de C. Sin embargo, para instancias particulares del código existen buenos algoritmos de decodificación que permite realizar esta eliminación del error de manera eficiente, es decir con complejidad polinómica.

La idea de McEliece es utilizar uno de estos códigos fácilmente descodificables como clave privada, pero disfrazarlo para presentar como clave pública un código general, a fin de enfrentar al criptoanalista con un problema computacionalmente imposible.

El criptosistema fue propuesto por Robert J. McEliece en 1978 [McEliece, 1978]. Este criptosistema está basado en la utilización de códigos lineales correctores de errores, utilizando originalmente códigos binarios de Goppa para encriptar y desencriptar mensajes.

Estos códigos son fáciles de construir y manejar para valores requeridos y poseen un buen algoritmo de decodificación. La propuesta original de McEliece utiliza $n \approx 1000$ y $k \approx 500$.

4.1.1. Códigos de Goopa

El código de Goppa $\Gamma(L, g(z))$ es definido por el polinomio de Goppa $g(z)$, este es un polinomio de grado t con coeficientes en el campo finito $\mathbf{F}_q^m$, para un número primo q y un subconjunto L de $\mathbf{F}_q^m$.

$$g(z) = g_0 + g_1 z + \cdots + g_t z^t = \sum_{i=0}^{t} g_i z^i,$$

$$L = \{\alpha_1, \alpha_2, \ldots, \alpha_m\} \subseteq \mathbf{F}_q^m,$$

tal que $g(\alpha_i) \neq 0$ para todo $\alpha_i \in L$. Para un vector $c = (c_1, c_2, \ldots, c_n)$ sobre $\mathbf{F}_q$ se le asocia la función

$$R_c(z) = \sum_{i=0}^{n} \frac{c_i}{z - \alpha_i},$$

donde $\frac{1}{z-\alpha_i}$ es el único polinomio para el cual $(z - \alpha_i).\dfrac{1}{z - \alpha_i} \equiv 1 \pmod{g(z)}$ con un grado menor que o igual a $t - 1$.

Definición 4.1 El código de Goppa $\Gamma(L, g(z))$ consiste de todos los vectores c tales que

$$R_c(z) \equiv 0 \pmod{g(z)},$$

esto es que $g(z)|R_c(z)$.

Recordemos que en un $[n, k, d]$ código lineal, este tiene tamaño n, dimensión k y distancia mínima d.

Teorema 4.2 El código de Goppa $\Gamma(L, g(z))$ de tamaño n es un código lineal sobre $\mathbf{F}_q$ con las propiedades

- la dimensión del código satisface que $k \geq n - mt$,
- la distancia mínima del código satisface $d \geq t + 1$.

Prueba: Ya que $\frac{1}{z-\alpha_i}$ se puede ver como un polinomio $p_i(z)$ módulo $g(z)$.

$$\frac{1}{z - \alpha_i} \equiv p_i(z) = p_{i1} + p_{i2}z + \cdots + p_{it}z^{t-1} \pmod{g(z)}.$$

Además, la ecuación de la Definición 4.1 se puede reescribir como

$$\sum_{i=1}^{n} c_i p_i(z) \equiv 0 \pmod{g(z)},$$

o si se toma los coeficientes de z^j separadamente como

$$\sum_{i=1}^{n} c_i p_{ij} = 0, \text{ para } 1 \leq j \leq t.$$

Esto significa que $\Gamma(L, g(z))$ puede ser definido como t ecuaciones lineales con coeficientes en $\mathbf{F}_q^m$, que se reduce al menos a mt ecuaciones lineales sobre $\mathbf{F}_q$. En consecuencia, la dimensión k de $\Gamma(L, g(z))$ debe ser al menos $n - mt$.

Sabiendo que el código es lineal, recordemos que en un código lineal, la distancia mínima es igual al peso mínimo de una palabra no nula [Gómez and Tena, 1997]. Para la segunda propiedad, se asume que c es una palabra no nula de peso w y $c_i \neq 0$ para $i \in \{i_1, i_2, \ldots, i_w\}$. Por ello, reescribimos $R_c(z)$ como

$$\sum_{i=1}^{n} \frac{c_i}{z - \alpha_i} = \frac{\sum_{j=1}^{w} c_{i_j} \prod_{1 \leq k \leq w, k \neq j} (z - \alpha_i)}{\prod_{j=1}^{w} (z - \alpha_i)}.$$

Ya que el denominador no tiene factores en común con $g(z)$, $g(z)$ debe ser un divisor del numerador, se debe cumplir para la ecuación en la Definición 4.1. El numerador tiene grado menor que o igual a $w - 1$. Entonces, $w - 1 \geq t$ y la distancia mínima $d = w \geq t + 1$. □

Definición 4.3 Si n, m, t son enteros positivos y

$$g(z) = \sum_{i=0}^{t} g_i z^i \in \mathbf{F}_{2^m}[z]$$

es un polinomio mónico de grado t. Además, $L = (a_1, a_2, \ldots, a_n) \in \mathbf{F}_{2^m}^n$ es una tupla de n elementos distintos de $\mathbf{F}_{2^m}$ tales que

$$g(a_i) \neq 0, \text{ para todo } i, 1 \leq i \leq n.$$

El código *binario de Goppa* $\mathcal{G}(L, g(z))$ consiste de todos los elementos $c = (c_1, c_2, \ldots, c_n) \in \{0,1\}^n$ que satisfacen

$$\sum_{i=1}^{n} \frac{c_1}{z - a_i} \equiv 0 \pmod{g(z)}.$$

Definición 4.4 Si el polinomio $g(z)$ de grado t con coeficientes en el grupo finito $\mathbf{F}_{2^m}$ no tiene raíces de multiplicidad mayor que 1, recibe el nombre de polinomio separable.

Proposición 4.5 Sea $\mathcal{G}(L, g(z))$ un código binario de Goppa con un polinomio separable de grado t. Entonces la distancia mínima d del código $\mathcal{G}(L, g(z))$ es al menos $2t + 1$.

Prueba: Si $L = \{a_1, a_2, \ldots, a_n\}$ y c es una palabra no nula con peso w. Sabemos de la prueba del Teorema 4.2 que

$$\sum_{i=1}^{n} \frac{c_i}{z - a_i} \equiv \pmod{g(z)} \iff g(z) | f(z),$$

en el cual

$$f(z) = \sum_{j=1}^{w} c_{i_j} \prod_{1 \leq k \leq w, k \neq j} (z - a_i) = \sum_{j=1}^{w} \prod_{1 \leq k \leq w, k \neq j} (z - a_{i_j}),$$

ya que el código es binario. Además, la última expresión es la derivada de la función $\prod_{j=1}^{w}(z - a_{i_j})$ y como una derivada binaria solo puede tener términos de coeficientes par, en consecuencia

$$f(z) = f_0 + f_2 z^2 + \cdots + f_{2u} z^{2u}, \text{ con } 2u \leq w - 1.$$

En $\mathbf{F}_2^m$ esto es equivalente a

$$f(z) = (k_0 + k_2 z + \ldots k_{2u} z^u)^2, \text{ con } k_i^2 = f_i \text{ y } 2u \leq w - 1.$$

Entonces $g(z)$ divide a $(k(z))^2$, con $k(z)$ un polinomio de grado u y $2u \leq w - 1$. Ya que $g(z)$ no tiene raíces dobles, $g(z)$ también divide a $k(z)$. Finalmente $t \leq u$ y $w - 1 \geq 2u \geq 2t$. Por tanto, la distancia mínima d es al menos $2t + 1$. □

Teniendo en cuenta estas definiciones y la escritura anterior de $\frac{1}{z - a_i}$, se obtiene la siguiente matriz

$$\left(\frac{1}{g(a_0)} \frac{g(z) - g(a_0)}{z - a_0}, \frac{1}{g(a_1)} \frac{g(z) - g(a_1)}{z - a_1} \ldots, \frac{1}{g(a_{n-1})} \frac{g(z) - g(a_{n-1})}{z - a_{n-1}} \right),$$

donde cada término polinómico debe de interpretarse como un vector columna, esto es, si se denota $g(z) = g_0 + g_1 z + \cdots + g_t z^t$ y $h_i = g(a_1)^{-1}$ se obtiene

$$\frac{g(z) - g(a_i)}{z - a_i} = g_t(z^{t-1} + z^{t-1} a_i + \cdots + a_i^{t-1}) + \cdots + g_2(z - a_i) + g_1;$$

con esto y escritos los coeficientes como vectores columnas se obtiene la siguiente matriz

$$\begin{pmatrix} h_0 g_t & \ldots & h_{n-1 g_t} \\ h_0(g_{t-1} + g_t a_0) & \ldots & h_{n-1}(g_{t-1} + g_t a_{n-1}) \\ \vdots & & \vdots \\ h_0(g_1 + g_2 a_0 + \cdots + g_t a_0^{t-1}) & \ldots & h_{n-1}(g_1 + g_2 a_{n-1} + \cdots + g_t a_{n-1}^{t-1}) \end{pmatrix}$$

y esta matriz se puede factorizar como

$$\begin{pmatrix} g_t & 0 & \dots & 0 \\ g_{t-1} & g_t \dots & 0 & \\ \vdots & \vdots & \ddots & \vdots \\ g_1 & g_2 & \dots & g_t \end{pmatrix} \begin{pmatrix} h_0 & \dots & h_{n-1} \\ h_0 a_0 & \dots & h_{n-1} a_{n-1} \\ \vdots & & \vdots \\ h_0 a_0^{t-1} & \dots & h_{n-1} a_{n-1}^{t-1} \end{pmatrix}.$$

Se toma a la segunda de estas matrices como la matriz de control del código de Goppa, que además si en esta matriz sustituimos cada término por un vector columna de tamaño m con coeficientes en $\mathbf{F}_q$ tendremos una matriz con mt filas, no todas necesariamente independientes. Por ello, se obtiene el siguiente resultado.

Teorema 4.6 La dimensión de un código de Goppa es al menos $n - mt$.

4.1.2. Construcción original del criptosistema McEliece

La clave secreta contiene la descripción de la estructura del código lineal, seleccionado un algoritmo de generación de la llave y la clave pública es tomada como una versión *aleatorizada* del mismo código, el cual es difícil distinguir de un código lineal aleatorio. Desencriptar requiere un algoritmo eficiente para el código lineal seleccionado, intuitivamente, se conoce la estructura del código lineal fundamental (clave secreta) que proporciona una función trampa para la rápida desencriptación, pero es difícil de realizar sin este conocimiento.

La construcción original usa códigos binarios de Goppa, los cuales son adecuados para aplicaciones criptográficas debido a sus altas capacidades correctoras de errores y a que usan una matriz generadora densa, la cual es difícil de distinguir de una matriz binaria aleatoria.

Definición 4.7 Una matriz de permutación es una matriz binaria P de tamaño $n \times n$ donde cada columna y cada fila contienen un único 1 y el resto de elementos son ceros.

Multiplicar cualquier matriz A de tamaño $k \times n$ con una matriz de permutaciones P resulta en una matriz $A' = AP$ que contiene las mismas columnas de A, pero orden permutado.

Generación de la clave

Describimos a continuación como generar la clave de los criptosistemas originales de McEliece:

- Elegir un código lineal $[n, k, 2t + 1]$ aleatorio C sobre $\mathbf{F}_2$ que posea un algoritmo eficiente de decodificación D que pueda corregir hasta t errores.
- Calcular una $k \times n$ matriz generadora G de C.
- Generar una $k \times k$ matriz aleatoria binaria S, invertible.
- Generar una $n \times n$ matriz de permutación aleatoria P
- Calcular la $k \times n$ matriz $G' = SGP$.
  A partir de lo anterior,
  - La clave pública es (G', t).
  - La clave privada es (S, G, P, D).

Veamos a continuación, como cifrar y descifrar.

Encriptación

Sea un texto plano $m \in \{0, 1\}^k$. Seleccionar un vector aleatorio $e \in \{0, 1\}^n$ de peso t y calcular el texto cifrado como

$$c = mG' + e$$

Desencriptación

Dado el texto cifrado $c \in \{0,1\}^n$, primero calcula

$$cP^{-1} = (mS)G + eP^{-1}.$$

Ya que

$$(mS)G$$

es una palabra válida del código lineal seleccionado y eP^{-1} tiene peso t, el algoritmo de decodificación D puede ser aplicado a cP^{-1} para obtener $c' = mS$. Entonces calcular m con

$$m = c'S^{-1}$$

4.1.3. Seguridad del criptosistema de McEliece

En el caso del método original de McEliece, se toma como G, un código de Goppa $\mathcal{G}(a_1, a_2, \ldots, a_n, g(X)) \subseteq \mathbf{F}_2^n$ con $g(X) \in \mathbf{F}_{2^m}[X]$ y $a_i \in \mathbf{F}_{2^m}$, capaz de corregir a lo más t errores. La dimensión del código de Goppa para aplicaciones criptográficas se toma como $k = n - tm$. Sea G una $k \times n$ matriz generadora de $\mathcal{G}$ y $G' = SGP$ una clave pública de criptosistema de McEliece con $k \times k$ matriz de permutación binaria no singular P.

La seguridad del criptosistema de McEliece viene sugerida por los siguientes problemas de decisión en la teoría de códigos.

Problema 4.8 Sea C un $[n,k]$ código lineal sobre $\mathbf{F}$ y $a \in \mathbf{F}^n$. Encontrar una palabra $c \in C$ tal que la distancia $d(a,c)$ es mínima.

Problema 4.9 Sea C un $[n,k]$ código lineal sobre un campo $\mathbf{F}$ y $w \in \mathbf{N}$. Encontrar una palabra $c \in C$ tal que su peso $\mathrm{wt}(c) = w$.

En [Berlekamp et al., 2006] se prueba que estos dos problemas son NP-Completos ya que los mejores algoritmos conocidos para tratar de resolverlos toman tiempo exponencial, por lo que para valores grandes en los parámetros del código resolver los problemas no es computacionalmente factible.

El hecho que estos dos problemas sean NP-Completos sugiere indicar que romper el criptosistema de McEliece sea también NP-Completo ya que los códigos irreducibles binarios de Goppa solo son una fracción de todos los posibles códigos lineales.

4.1.4. Código dual de McEliece: criptositema de Niederreirter

Esta variante del criptosistema de McEliece fue propuesta por H. Niederreirter en 1968. Este criptosistema es muy similar al de McEliece pero usa una matriz de paridad en lugar de una matriz de generación. Además, el sistema está basado en la idea del algoritmo de decodificación por síndrome.

Generación de la clave

Para generar la clave del criptosistema se siguen los siguientes puntos:

- Seleccionar un $[n, k, 2t+1]$ código lineal aleatorio C sobre $\mathbf{F}_2$ que tenga un algoritmo eficiente de decodificación por síndrome D que puede corregir a lo más t errores.
- Calcular una $(n-k) \times n$ matriz de paridad H de C.
- Generar una $(n-k) \times (n-k)$ matriz binaria S no singular.
- Generar una $n \times n$ matriz de permutación P.
- Calcular una $(n-k) \times n$ matriz $H' = SHP$.
  Entonces tomamos como

- clave pública a (H', t).
- clave privada a (S, H, P, D).

A continuación indicamos el proceso de encriptación y desencriptación.

Encriptación

Para encriptar un texto plano $m \in \{0,1\}^k$ con peso t, calcular el texto cifrado como el síndrome de m:

$$c = mH'^T.$$

Desencriptación

Para desencriptar el texto cifrado c, calcular

$$S^{-1}c^T = HPm^T$$

y a continuación encontrar un vector $z \in \mathbf{F}^n$ tal que $Hz^T = HPm^T$. Entonces $z-(Pm^T)^T = z-mP^T$ es una palabra válida en C. Como mP^T tiene peso t, se puede aplicar el algoritmo eficiente de decodificación D sobre z para encontrar el vector de errores mP^T y por ello m.

En [Li et al., 2006] se muestra que el criptosistema de McEliece y el criptosistema de Niederreiter son equivalentes, esto es, si alguien rompe la seguridad de uno de ellos, tambien lo habrá hecho con la seguridad del otro. Esto se tiene ya que McEliece y Niederreiter usan códigos lineales que son duales uno del otro y una matriz generadora pude ser calculada de una matriz de paridad y viceversa.

Nuestra siguiente sección trata sobre posibles ataques a los criptosistemas antes mencionados.

4.1.5. Ataques al Criptosistema de McEliece

Indicamos cuatro posibles ataques en esta sección.

Ataque contra la clave privada

El ataque más conocido de recuperación de la clave secreta a partir de la clave pública fue propuesto por Loidreau y Sendier en 2001 [Loidreau and Sendrier, 2001]. El ataque presentado solo es factible cuando se usa una clave débil específica, es decir, cuando el polinomio de Goppa tiene coeficientes binarios ($\mathbf{F}_2$ en lugar de $\mathbf{F}_{2^m}$).

La idea del ataque es utilizar el algoritmo de división de soporte propuesto por Sendier, que permite calcular la permutación entre dos códigos lineales equivalentes por permutación. El ataque general busca exhaustivamente un código de permutación equivalente con la clave pública McEliece entre todos los posibles códigos Goppa para encontrar la clave secreta.

Para el caso específico donde los coeficientes del polinomio de Goppa son de $\mathbf{F}_2$, el ataque se puede hacer mucho más rápido, pero aún se requiere una gran cantidad de cálculos para romper una sola clave.

En el caso general, este ataque se convierte rápidamente en inviable para cualquier opción razonable de parámetros para el código Goppa, ya que se necesitan ser enumerados aproximadamente $2^{\frac{m(t-3)}{mt}}$ códigos Goppa y el algoritmo de división de apoyo tiene que ser ejecutado en cada iteración.

Para los parámetros originales de McEliece $n = 1024$ y $k = 512$ esto se obtiene aproximadamente en 2^{461} pasos.

Ataque de decodificación de conjuntos de información.

Este ataque ya fue presentado por McEliece [McEliece, 1978] en su artículo original del criptosistema y mejorado por Lee y Brickell en 1988 [Lee and Brickell, 1988]. El ataque propuesto es un algoritmo general para decodificar cualquier código lineal de corrección de errores por lo que resuelve el problema de decodificación general NP-hard (problema 1). El ataque tiene complejidad computacional exponencial, sin embargo, es útil analizarlo ya que la misma idea general se usa en posteriores ataques más eficientes.

El ataque se basa en el método de decodificación de conjuntos de información. Sea C un $[n, k, 2t+1]$ código lineal sobre el campo $\mathbf{F}$ y G su matriz generadora. Se supone que el criptoanalista no conoce un algoritmo de decodificación eficiente para C (es decir, C es un código Goppa binario y el polinomio generador es desconocido). Si c es un texto cifrado con el criptosistema de McEliece, $c = mG + e$, donde $m \in \mathbf{F}^k, e \in \mathbf{F}^n$ y $\mathrm{wt}(e) = t$. Escribir J_I la matriz que contiene solo columnas en los índice $I \subseteq \{1, 2, \ldots, n\}$ de G. Para desencriptar c el criptoanalista puede hacer los siguiente:

1. Seleccionar k índice $I \subset \{1, 2, \ldots, n\}, |I| = k$, con la esperanza de que no exista error en c con esos índices.

2. Entonces se cumple la siguiente relación $c_I = mG_I + e_I$, si $\mathrm{wt}(I) = 0$. Entonces el criptoanalista puede encontrar m al calcular $m = c_I G_I^{-1}$.

3. El criptoanalista verifica $\mathrm{wt}(c_I G_I^{-1} G + c) = t$, si este es el caso, entonces se calcula $m = c_I G_I^{-1}$, en otro caso, vuelve al paso 1.

4. El criptoanalista verifica que $\mathrm{wt}(c_I G_I^{-1} G + c) = t$, si este es el caso entonces se calcula $m = c_I G_I^{-1}$, en otro caso vuelve al paso 1.

El trabajo realizado se espera que sea aproximadamente de tamaño,

$$W = k^3 \frac{\binom{n}{k}}{\binom{n-t}{k}},$$

ya que la probabilidad de no tener errores en los k índices elegidos es $\frac{\binom{n-t}{k}}{\binom{n}{k}}$ y la matriz de inversión tiene complejidad aproximadamente $O(k^3)$ [Loidreau and Sendrier, 2001].

Este ataque puede mejorarse tratando de encontrar el vector de errores correcto e_I con peso $\mathrm{wt}(e_I) \leq j$, para algún $j < t$. El algoritmo mejorado será entonces

1. El criptoanalista fija un $j < t$ y selecciona k índices $I \subset \{1, 2, \ldots, n\}, |I| = k$.

2. Se cumple la siguiente relación $c_I = mG_I + e_I$. El criptoanalista calcula $Q = G_I^{-1} G$.

3. Para todos los vectores e_I con peso $\mathrm{wt}(e_I) \in \{0, 1, \ldots, j\}$, el criptoanalista calcula $e' = c + c_I Q + e_I Q$ ya que $c_I Q = mG + e_I Q$.

4. Si $\mathrm{wt}(e') = t$, el criptoanalista obtiene $m = (c_I + e_I) G_I^{-1}$. En otro caso continua nuevamente desde la etapa 3. Si todos los e_I no son satisfactorios el criptoanalista retorna al paso 1.

El número esperado de intentos para elegir I de modo que haya como máximo j errores en c_I es

$$T_j = \frac{\binom{n}{k}}{\sum_{i=0}^{j} \binom{t}{i}\binom{n-t}{k-i}}.$$

El número de vectores error e_I con $\mathrm{wt}(e_I) \leq j$ es

$$N_j = \sum_{i=1}^{j} \binom{k}{i}.$$

Por lo que el factor de trabajo esperado al implementar el algoritmo es

$$W_j = T_j(k^3 + kN_j).$$

Ataques para encontrar palabras de bajo peso

Basados en la idea de decodificación de conjuntos de información, existen ataques más eficaces que los anteriores. Encontrar el texto llano x a partir de un texto cifrado $y \in \mathbf{F}_{2^n}$ encriptado con un código lineal C puede reducirse a la búsqueda de una palabra de código de peso pequeños en un código lineal ligeramente mayor como lo mencionan Canteaut y Chabaud en su artículo [Canteaut and Chabaud, 2006]. Esta reducción relaciona el ataque con el problema NP-hard de encontrar palabras de código de peso específico. Este ataque no es específico de los códigos Goppa, sino que puede aplicarse a cualquier código lineal de corrección de errores.

Sea C un $[n, k, 2t + 1]$ código lineal sobre $\mathbf{F}$. Sea $y \in \mathbf{F}_{2^n}$ una palabra donde $c \in C$ es la palabra más cercana con $d(y, c) = t$, se puede mostrar que $y + c$ es la única palabra de peso t en un código extendido $C + \{y\}$, donde $C + \{y\}$ significa que el vector y es agregado a la matriz generadora de C como una nueva columna.

Ya que la distancia mínima de C es $2t + 1$, entonces y no puede ser una palabra de C, por lo que la matriz generadora de C con y como agregado como fila es un nuevo código lineal $C' = \{y + x | x \in C\}$ con dimensión $k + 1$ y además, $c + y \in C'$.

La palabra c debe ser la única palabra en C a distancia t de y, ya que la distancia mínima de C es $2t+1$. Por tanto la palabra $y+c$ es de hecho, la única palabra en C' con peso t. Si el criptoanalista puede encontrar esta palabra, puede obtener el texto enviado.

El ataque más conocido de este tipo lo presentó *Bernstein* en 2008 en su artículo [Bernstein et al., 2008]. Sin embargo todos los ataques avanzados de decodificación de conjuntos de información de este tipo tienen un costo de $c^{\frac{(1+o(1))n}{\log(n)}}$, donde $c = \dfrac{1}{(1+R)^{1-R}}$ y R es la ratio del código lineal $\frac{k}{n}$.

Este ataque se ejecuta en tiempo exponencial a medida que el tamaño del código lineal aumenta.

Recientemente para códigos geométrico algebraicos se ha producido un gran avance.

Ataque basado en códigos geométrico algebraicos y en subcódigos

Este ataque fue presentado en 2017 [Couvreur et al., 2017] y presenta un ataque en tiempo polinómico, ya sea a códigos geométrico algebraicos (AG) o en pequeños subcódigos codimensionales de códigos AG. Estos ataques consisten en la reconstrucción a ciegas de un par de corrección de errores (ECP) o una matriz de corrección de errores (ECA) a partir de los datos individuales de una matriz generadora arbitraria de un código.

Para un código público de longitud n sobre $\mathbf{F}_q$, estos ataques se ejecutan en $O(n^4 \log(n))$ operaciones en $\mathbf{F}_q$ para la reconstrucción de un ECP y $O(n^5)$ operaciones para la reconstrucción de un ECA.

4.1.6. Variantes del Criptosistema de McEliece

Un impedimento que tiene la implementación del criptosistema de McEliece es el gran tamaño de su clave pública, esto ha motivado a la creación de diversas variantes de este criptosistema al utilizar diferentes familias de códigos lineales con el afán de reducir el tamaño de la clave pública.

Códigos binarios de Goppa

Esta es la familia de códigos lineales propuesta por McEliece. En la tabla siguiente se hace una comparativa del tamaño de la clave pública entre el sistema de McEliece y el sistema RSA, donde podemos notar claramente una desventaja del sistema de McEliece.

Nivel seguridad(bit)	$[n, k]$	t	Tamaño de la clave(bits)	RSA tamaño de la clave (bits)
80	[1632,1269]	33	460647	512
128	[2960,2280]	56	1537536	3072
256	[6624,5129]	115	7667855	15360

Códigos generalizados de Reed Solomon

Esta familia de códigos fue propuesta para ser usada en el criptosistema de Niederreiter pero se demostró que era inseguro seis años después de la implementación por Sidelnikov y Shestakov al proponer un ataque en tiempo polinomial [Sidelnikov and Shestakov, 1992], aunque la idea de utilizar códigos generalizados de Reed Solomon sigue siendo aún de gran interés para la comunidad científica.

Códigos de Reed Muller

Esta familia de códigos fue propuesta por *Sidelnikov* [Sidelnikov, 1994]. Estos códigos han recibido ataques [Minder and Shokrollahi, 2007], al utilizar un tipo de ataque de filtración basado en la estructura de la palabra que proporciona la distancia mínima. Recientemente [Borodin and Chizhov, 2014], se propuso otro ataque que podría resolver el problema de equivalencia de código para algunos de los parámetros de los códigos de Reed Solomon en tiempo polinomial.

Códigos salvajes de Goppa

Esta familia de códigos es una extensión natural de los códigos binarios de Goppa a un campo no binario, [Bernstein et al., 2011]. Fue quebrado en 2014 utilizando un tipo de técnica de filtración [Couvreur et al., 2013].

Códigos de Srivastava

Los códigos de Srivastava fueron propuestos en 2012 para reducir la clave pública del sistema original de McEliece usando códigos quasi-diádicos de Srivastava que también sirven para generar esquemas de firmas. En 2016 fue quebrado el sistema de firmas pero el esquema de encriptación aún sigue siendo válido.

Nivel de seguridad (bits)	Códigos sálvajes de Goopa	Códigos de Srivastava	Códigos Binarios de Goopa
80	-	36288	460647
128	1523278	37440	1537536

Códigos Geométrico Algebraicos

Esta familia de códigos fue propuesta en 1996 [Janwa and Moreno, 1996] y en 2017 se propuso un ataque en tiempo polinomial que trabaja sobre curvas de género arbitrario [Couvreur et al., 2017]. Los autores de este ataque mencionan que está fuera del alcance de su estudio el caso de subcódigos de códigos geométrico algebraicos de subcampos con género cero. Por ello **son posibles candidatos para una generalización del esquema de McEliece**.

Códigos de evaluación correctores

Se han estudiado en el capítulo 2, donde se demuestra que puede tener ventaja su utilización en el criptosistema de McEliece con respecto a la cantidad de errores que se pueden corregir y al algoritmo eficiente a utilizar. Este trabajo pretende ser la etapa germinal de un futuro estudio del esquema de McEliece con este código como soporte del mismo.

4.2. McEliece y la criptografía post-cuántica

En 1996 aparece **el algoritmo de Grover** que es una transformación de los circuitos convencionales a los circuitos cuánticos. La entrada a la transformación es un circuito que calcula una función $f : \mathbf{F}_2^b \to \mathbf{F}_2$ y la salida es un circuito cuántico que calcula una raíz de f (si esta existe), esto es una cadena de b-bits tal que $f(x) = 0$ [Grover, 1996].

Este algoritmo puede encontrar una clave de 256 bits del sistema AES utilizando cerca de 2^{128} operaciones cuánticas, dando con el texto original a partir de la clave [Bernstein, 2010].

En el 2010 , Bernstein prueba que el criptosistema de McEliece es seguro contra ataques post-quantum [Bernstein, 2010], tales como el algoritmo de Grover, aunque indica que para lograrlo el tamaño de la clave debería de cuadriplicarse.

Por último mencionamos que el criptosistema de McEliece es inmune al algoritmo de Shor, el cual se ha mencionado en el primer capítulo es capaz de quebrar los criptosistemas actuales.

4.3. Conclusiones

Los sistemas criptográficos más populares de la actualidad dejarán de ser seguros una vez que la computadora cuántica ya no sea un reto para los ingenieros actuales y empiecen a construirse. Este evento es posible que ocurra en los próximos años. Por ello que debemos de prepararnos ya que sus repercusiones podrían llegar a ser catastróficas.

La criptografía basada en códigos presenta una posible solución a este problema, pues contiene diferentes familias de códigos que pueden ser implementados en el sistema de criptografía de McEliece.

Los códigos lineales de evaluación correctores se pueden utilizar en el criptosistema de McEliece ya que tienen algoritmos de decodificación eficientes y la cantidad de errores a corregir es considerada buena.

El criptosistema de McEliece presenta desventajas respecto a otros criptosistemas por el gran tamaño de su clave, pero presenta la ventaja que es un criptosistema resistente a los ataques actuales, inclusive a los ataques por computador cuántico.

La intersección entre la criptografía y la teoría de códigos es aún un campo en exploración que puede generar aún más lineas de investigación que den solución a los actuales y a los siguientes retos de la humanidad en el área de la seguridad informática.

Capítulo 5

Anexos

Algoritmo 5.1: Algoritmo del símbolo de legendre

Input: $a, b \in \mathbb{Z}^+$

if $b \pmod 2 = 0$ **then**
  return 0
$j \leftarrow 1$
if $a < 0$ **then**
  $a \leftarrow -a$
  if $b \pmod 4 = 3$ **then**
    $j \leftarrow -j$
while $a \neq 0$ **do**
  while $a \pmod 2 = 0$ **do**
    $a \leftarrow \frac{a}{2}$
    if $b \pmod 8 = 3 \vee b \pmod 8 = 5$ **then**
      $j \leftarrow -j$
  temp $\leftarrow a$
  $a \leftarrow b$
  $b \leftarrow$ temp
  if $a \pmod 4 \wedge b \pmod 4 = 3$ **then**
    $j \leftarrow -j$
    $a \leftarrow a \pmod b$
if $b = 1$ **then**
  return j
else
  return 0

Algoritmo 5.2: Square And Multiply

Input: x, c, n

$z \leftarrow 1$
for $i \leftarrow l - 1$ **to** 0 **do**
  $z \leftarrow z^2 \pmod n$
  if $c_i = 0$ **then**
    $z \leftarrow (z \times x) \pmod n$
return z

Algoritmo 5.3: Algoritmo extendido de Euclides

Input: $a, b \in \mathbb{N}$
Output: r, s, t
Result: $r = sa + tb$

$a_0 \leftarrow a$
$b_0 \leftarrow b$
$t_0 \leftarrow 0$
$t \leftarrow 1$
$s_0 \leftarrow 1$
$s \leftarrow 0$
$q \leftarrow \left\lfloor \frac{a_0}{b_0} \right\rfloor$
$r \leftarrow a_0 - qb_0$
while $r > 0$ **do**
    temp $\leftarrow t_0 - qt$
    $t_0 \leftarrow t$
    $t \leftarrow$ temp
    temp $\leftarrow s_0 - qs$
    $s_0 \leftarrow s$
    $s \leftarrow$ temp
    $a_0 \leftarrow b_0$
    $b_0 \leftarrow r$
    $q \leftarrow \left\lfloor \frac{a_0}{b_0} \right\rfloor$
    $r \leftarrow a_0 - qb_0$
$r \leftarrow b_0$
return (r, s, t)

Algoritmo 5.4: Multiplicativo Inverso

Input: a, b

$a_0 \leftarrow a$
$b_0 \leftarrow b$
$t_0 \leftarrow 0$
$t \leftarrow 1$
$q \leftarrow \left\lfloor \frac{a_0}{b_0} \right\rfloor$
$r \leftarrow a_0 - qb_0$
while $r > 0$ **do**
    temp $\leftarrow (t_0 - qt) \pmod{a}$
    $t_0 \leftarrow t$
    $t \leftarrow$ temp
    $a_0 \leftarrow b_0$
    $b_0 \leftarrow r$
    $q \leftarrow \left\lfloor \frac{a_0}{b_0} \right\rfloor$
    $r \leftarrow a_0 - qb_0$
if $b_0 \neq 1$ **then**
    no tiene inverso módulo a
else
    return (t)

Algoritmo 5.5: P.DECOMPRESS

Input: x, i

$z \leftarrow x^3 + ax + b \pmod{p}$

if *z es residuo cuadrático módulo p* **then**

  return Fallo

else

  $y \leftarrow \sqrt{z} \pmod{p}$

  if $y \equiv i \pmod{2}$ **then**

    return (x, y)

  else

    return $(x, p - y)$

Algoritmo 5.6: P.COMPRESS

Input: $(x, y) \in E$

return $(x, y \pmod{2})$

Algoritmo 5.7: Algoritmo de Shanks

Input: n, α, β

$m \leftarrow \lceil \sqrt{n} \rceil$

for $j \leftarrow 0$**to** $m - 1$ **do**

  Calcular α^{mj}

Ordenar los m pares ordenados (j, α^{mj}) con respecto a su segunda coordenada obteniendo una lista L_1

for $i \leftarrow 0$**to** $m - 1$ **do**

  Calcular $\beta\alpha^{-i}$

Ordenar los m pares ordenados $(i, \beta\alpha^{-i})$ con respecto a sus segunda coordenada, obteniendo una lista L_2

Encontrar el par ordenado $(j, y) \in L_1$ y el par ordenado $(i, y) \in L_2$

$\log_\alpha \beta \leftarrow (mj + i) \pmod{n}$

Algoritmo 5.8: Algoritmo Pollar Rho para el logaritmo discreto

Input: G, n, α, β
if $x \in S_1$ **then**
  $f \leftarrow (\beta.x, a, (b+1) \pmod n)$
else
  if $x \in S_2$ **then**
    $f \leftarrow (x^2, 2a \pmod n, 2b \pmod n)$
    else
      $f \leftarrow (\alpha.x, (a+1) \pmod n, b)$
return (f)
Definir la partición $G = S_1 \cup S_2 \cup S_3$
$(x, a, b) \leftarrow f(1, 0, 0)$
$(x', a', b') \leftarrow f(x, a, b)$
while $x \neq x'$ **do**
  $(x, a, b) \leftarrow f(x, a, b)$
  $(x', a', b') \leftarrow f(x', a', b')$
  $(x', a', b') \leftarrow f(x', a', b')$
if $\gcd(b' - b, n) \neq 1$ **then**
  return (**FALLA**)
else
  return $((a - a')(b' - b)^{-1} \pmod{()n})$

Algoritmo 5.9: Algoritmo de Pohlig-Hellman

Input: $G, n, \beta, \alpha, q, c$
$j \leftarrow 0$ $\beta_j \leftarrow \beta$
while $j \leq c - 1$ **do**
  $\delta \leftarrow \beta_j^{n/q^{j+1}}$
  Encontrar i tal que $\delta = \alpha^{in/q}$
  $a_j \leftarrow i$
  $\beta_{j+1} \leftarrow \beta_j \alpha^{-a_j q^j}$
  $j \leftarrow j + 1$
return $(a_0, \ldots, a_{c-1})$

Bibliografía

[Bach and Shallit, 1996] Bach, E. and Shallit, J. (1996). *Algorithmic Number Theory*. MIT Press, Cambridge, MA, USA.

[Berlekamp et al., 2006] Berlekamp, E., McEliece, R., and van Tilborg, H. (2006). On the Inherent Intractability of Certain Coding Problems (Corresp.). *IEEE Trans. Inf. Theor.*, 24(3):384–386.

[Bernstein, 2010] Bernstein, D. J. (2010). Grover vs. mceliece. In *Proceedings of the Third International Conference on Post-Quantum Cryptography*, PQCrypto'10, pages 73–80, Berlin, Heidelberg. Springer-Verlag.

[Bernstein et al., 2008] Bernstein, D. J., Lange, T., and Peters, C. (2008). Attacking and Defending the McEliece Cryptosystem. In *Proceedings of the 2Nd International Workshop on Post-Quantum Cryptography*, PQCrypto '08, pages 31–46, Berlin, Heidelberg. Springer-Verlag.

[Bernstein et al., 2011] Bernstein, D. J., Lange, T., and Peters, C. (2011). Wild McEliece. In Biryukov, A., Gong, G., and Stinson, D. R., editors, *Selected Areas in Cryptography*, pages 143–158, Berlin, Heidelberg. Springer Berlin Heidelberg.

[Binet, 1841] Binet, J. P. M. (1841). Recherches sur la théorie des nombres entiers et sur la résolution de l'équation indéterminée du premier degré qui n'admet que des solutions entières. *Journal de Mathématiques Pures et Appliquées*, 6:449–494.

[Borodin and Chizhov, 2014] Borodin, M. A. and Chizhov, I. V. (2014). Effective attack on the McEliece cryptosystem based on Reed-Muller codes. volume 24, pages 273–280. Exported from https://app.dimensions.ai on 2018/09/20.

[Canteaut and Chabaud, 2006] Canteaut, A. and Chabaud, F. (2006). A New Algorithm for Finding Minimum-weight Words in a Linear Code: Application to McEliece's Cryptosystem and to Narrow-sense BCH Codes of Length 511. *IEEE Trans. Inf. Theor.*, 44(1):367–378.

[Conrad, 2017] Conrad, K. (2017). The Miller – Rabin Test.

[Cormen et al., 2009] Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2009). *Introduction to Algorithms, Third Edition*. The MIT Press, 3rd edition.

[Couvreur et al., 2017] Couvreur, A., Márquez-Corbella, I., and Pellikaan, R. (2017). Cryptanalysis of McEliece Cryptosystem Based on Algebraic Geometry Codes and Their Subcodes. *IEEE Trans. Inf. Theor.*, 63:5404–5418.

[Couvreur et al., 2013] Couvreur, A., Otmani, A., and Tillich, J. (2013). New Identities Relating Wild Goppa Codes. *CoRR*, abs/1310.3202.

[Gómez and Tena, 1997] Gómez, J. and Tena, J. (1997). *Codificación de la información*. Manuales y textos universitarios: Ciencias. Universidad de Valladolid.

[Grimaldi, 1998] Grimaldi, R. P. (1998). *Matemáticas discreta y combinatoria: introducción y aplicaciones*. Pearson Educación.

[Grover, 1996] Grover, L. K. (1996). A Fast Quantum Mechanical Algorithm for Database Search. In *Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing*, STOC '96, pages 212–219, New York, NY, USA. ACM.

[Hoholdt et al., 1998] Hoholdt, T., van Lint, J. H., and Pellikaan, R. (1998). Algebraic geometry codes. *The Handbook of coding theory*, 1:871–961.

[Janwa and Moreno, 1996] Janwa, H. and Moreno, O. (1996). McEliece public key cryptosystems using algebraic-geometric codes. *Designs, Codes and Cryptography*, 8(3):293–307.

[Koblitz, 1987] Koblitz, N. (1987). *A Course in Number Theory and Cryptography*. Springer-Verlag, Berlin, Heidelberg.

[Lee and Brickell, 1988] Lee, P. J. and Brickell, E. F. (1988). An Observation on the Security of McEliece's Public-key Cryptosystem. In *Lecture Notes in Computer Science on Advances in Cryptology-EUROCRYPT'88*, pages 275–280, New York, NY, USA. Springer-Verlag New York, Inc.

[Li et al., 2006] Li, Y. X., Deng, R. H., and Wang, X. M. (2006). On the Equivalence of McEliece's and Niederreiter's Public-key Cryptosystems. *IEEE Trans. Inf. Theor.*, 40(1):271–273.

[Loidreau and Sendrier, 2001] Loidreau, P. and Sendrier, N. (2001). Weak keys in the McEliece public-key cryptosystem. *IEEE Transactions on Information Theory*, 47(3):1207 – 1211.

[McEliece, 1978] McEliece, R. J. (1978). A public-key cryptosystem based on algebraic coding theory. *Deep Space Network Progress Report*, 42-44:114–116.

[Minder and Shokrollahi, 2007] Minder, L. and Shokrollahi, A. (2007). Cryptanalysis of the Sidelnikov Cryptosystem. In *Proceedings of the 26th Annual International Conference on Advances in Cryptology*, EUROCRYPT '07, pages 347–360, Berlin, Heidelberg. Springer-Verlag.

[Okeyinka, 2017] Okeyinka, A. E. (2017). Computational Complexity Study of RSA and ElGamal Algorithms. In Ao, S.-I., Kim, H. K., and Amouzegar, M. A., editors, *Transactions on Engineering Technologies*, pages 233–243, Singapore. Springer Singapore.

[Pommerening, 2016] Pommerening, K. (2016). The Euclidean Algorithm. https://www.staff.uni-mainz.de/pommeren/MathMisc/Euclid.pdf. Online; Acceso 1 de marzo del 2018.

[Shor, 1997] Shor, P. W. (1997). Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM J. Comput.*, 26(5):1484–1509.

[Sidelnikov, 1994] Sidelnikov, V. M. (1994). Public-key cryptosystem based on binary Reed-Muller codes. In *Discrete Mathematics and Applications*, volume 4, pages 191–208.

[Sidelnikov and Shestakov, 1992] Sidelnikov, V. M. and Shestakov, S. O. (1992). On insecurity of cryptosystems based on generalized Reed-Solomon codes. *Discrete Mathematics and Applications*, 2(4).

[Stinson, 2005] Stinson, D. (2005). *Cryptography: Theory and Practice, Third Edition*. Discrete Mathematics and Its Applications. Taylor & Francis.

[Yan, 2007] Yan, S. Y. (2007). *Cryptanalytic attacks on RSA*. Springer-Verlag, Berlin, Heidelberg.