Una breve introducción a la criptografía matemática

Aylen Martínez López y Martín Marfia

23 de noviembre de 2015

Índice

1. Introducción

La finalidad de la criptografía (de clave pública) es permitirle a dos personas intercambiar información confidencial, incluso si no se conocen y sólo pueden intercambiar datos a traves de un canal que puede ser monitoreado por un adversario. El principal objetivo de este trabajo es introducir una diversidad de tópicos matemáticos mientras simultáneamente se los integra con una descripción de la criptografía de clave pública moderna.

Por miles de años, todos los códigos y cifrados descansaron en el hecho de asumir que dos personas se intentan comunicar, en el trabajo los llaremos Bruno y Ana, compartiendo una clave secreta que su adversario, que llamaremos Inés, no posee. Bruno usará la clave para encriptar el mensaje, Ana la usará para decriptar el mensaje, e Inés sin la clave será incapaz de poder decriptar el mensaje. Una desventaja de este tipo de criptosistemas es que Bruno y Ana necesitan intercambiar la clave antes de poder transmitirse mensajes, pero a su vez permite que dos personas que nunca tuvieron un contacto directo intercambien mensajes cifrados.

A lo largo del trabajo veremos como la criptografía se basa en varias aréas de la matemática como la probabilidad, teoría de números, y estadística, pero en particular veremos como usa fuertemenete las estructuras de grupo, anillo y cuerpo. Por eso comenzaremos recordando y citando algunas definiciones y resultados de estructuras algebraicas. Y en los siguientes capítulos nos enfocaremos en el estudio del llamado Problema del Logaritmo Discreto y en el uso de la factorización entera como herramienta versátil en el desarrollo de algoritmos de encriptación.

2. Preliminares

En esta sección introductoria simplemente introduciremos algunos contenidos básicos de la teoría de grupos y anillos, como así también detallaremos algunos resultados que se realizan en el capítulo V del libro "Algebra"de Thomas Hungerford que serán utilizados a la mitad del trabajo para demostrar ciertas propiedades de los cuerpos finitos.

Para la parte de cuerpos finitos hemos añadido en cada resultado y definición la numeración usada en el libro para una mejor referencia al momento de usar estos resultados.

2.1. Grupos

En principio para la parte de teoría de grupos no vamos a detallar conceptos básicos como su definición, la de grupo conmutativo, grupo finito, u orden de un grupo. Pero si vamos a mirar algunos conceptos que van a ser clave:

Definición 2.1.1. Sea G un grupo,y $a \in G$ un elemento del grupo. Supongamos que existe un entero positivo d tal que $a^d = e$. Luego al mínimo tal d se lo llama el orden de a. Si no existe tal entero, se dice que el orden de a es infinito.

Proposición 2.1.2. *Sea G un grupo finito. Entonces cada elemento de G tiene orden finito. Más aún, si $a \in G$ tiene orden d y $a^k = e$, entonces $d|k$.*

Demostración. Como G es finito, la secuencia

$$a, a^2, a^3, a^4, ...$$

eventualemte se va a repetir, es decir, existe un entero positivo i y uno j tal que $a^i = a^j$. Multiplicando a ambos lados por a^{-i} y usando los axiomas de grupos tenemos que $a^{i-j} = e$. Como $i - j > 0$, probamos que alguna potencia de a es e. Sea d el mínimo entero positivo tal que $a^d = e$. Supongamos que $k \geq d$ también satisface que $a^k = e\cdot$. Usamos el TAD y tenemos que

$$k = dq + r \quad con\ 0 \leq r < d$$

Usando que $a^k = a^d = e$, tenemos que

$$e = a^k = a^{dq+r} = (a^d)^q \cdot a^r = e^q \cdot a^r = a^r$$

Pero d era el menor entero positivo tal que $a^d = e$, luego $r = 0$. Y así $k = dq$, luego $d \mid k$. ■

Proposición 2.1.3. *(Corolario del Teorema de Lagrange) Sea G un grupo finito y sea $a \in G$. Luego el orden de a divide al orden de G. Más precisamente, sea $n = |G|$ el orden de G y sea d el orden de a luego*

$$a^n = e \quad y \quad d|n.$$

Demostración. Para esta demostración primero vamos a definir el concepto de subgrupo generado: Dado un grupo G y $A \subseteq G$, decimos que el subgrupo generado por A en G es el menor subgrupo de G que contiene a A. Y lo notaremos $< A >$. Cuando $A = \{a\}$, con $a \in G$, $< A >$ está formado por todas las potencias distintas de a, luego $| < a > | = d$, donde d es el orden de a en G. El Teorema de Lagrange dice que el orden de todo subgrupo divide al orden del grupo. Aplicando este teorema a $< a >$ tenemos que $d|n$. Que $a^n = e$ es consecuencia de que $a^d = e$ y de que $d|n$. ■

2.2. Anillos

Para anillos no daremos su axiomática pero si daremos algunos resultados que valen en un anillo genérico y definiciones más necesarias:

Definición 2.2.1. Sean a y b elementos del anillo R con $b \neq 0$. Decimos que b divide a a, o que a es divisible por b, si existe un elemento $c \in R$ tal que $a = b \cdot c$. Y escribimos $b \mid a$ para indicar que b divide a a, y si b no divide a a escribimos $b \nmid a$

Proposición 2.2.2. *Sea R un anillo, entonces:*

1. *El elemento neutro $0 \in R$, i.e es el único elemento en R que satisface $0 + a = a + 0 = a$ para todo $a \in R$.*
2. *El neutro multiplicativo $1 \in R$ es único.*
3. *Todo elemento de R tiene un único inverso aditivo.*
4. *$0 \cdot a = a \cdot 0 = 0$ para todo $a \in R$*
5. *Notamos al opuesto de a por $-a$. Luego, $-(-a) = a$.*

6. *Sea* -1 *el opuesto del neutro multiplicativo* $1 \in R$. *Luego* $(-1) \cdot (-1) = 1$.

7. $b|0$ *para todo* $b \in R$, $b \neq 0$.

8. *Todo elemento de* R *tiene a lo sumo un inverso multiplicativo*

Demostración. 1. Supongamos que existen e_1 y e_2 que satisfacen $e_1 + a = a + e_1 = a$ y $e_2 + a = a + e_2 = a$ para todo $a \in R$. Entonces en particular,

$$e_1 = e_1 + e_2 = e_2$$

2. Supongamos que existen e_1 y e_2 que satisfacen $e_1 \cdot a = a \cdot e_1 = a$ y $e_2 \cdot a = a \cdot e_2 = a$ para todo $a \in R$. Entonces en particular,

$$e_1 = e_1 \cdot e_2 = e_2$$

3. Dado $a \in R$ supongamos que existen b y c que satisfacen $a + b = b + a = 0$ y $a + c = c + a = 0$. Entonces

$$b = 0 + b = c + a + b = c + 0 = c$$

4. $0 \cdot a = (0 + 0) \cdot a = (0 \cdot a) + (0 \cdot a)$ entonces sumando a ambos lados el opuesto de $a \cdot 0$, nos queda que $0 \cdot a = 0$.
Análogamente $a \cdot 0 = 0$

5. Notemos que $a + (-a) = -a + a = 0$ pues $-a$ es el opuesto de a. Entonces a es el opuesto de $-a$, esto es $-(-a) = a$.

6. Primero notemos que $(-1) \cdot a = a \cdot (-1) = -a$ ya que

$$(-1) \cdot a + a = (-1) \cdot a + 1 \cdot a = (-1 + 1) \cdot a = 0 \cdot a = 0, \text{ y}$$

$$a + (-1) \cdot a = 1 \cdot a + (-1) \cdot a = (1 - 1) \cdot a = 0 \cdot a = 0.$$

Entonces para todo $a \in R$,

$$(-1) \cdot (-1) \cdot a = (-1) \cdot (-a) = -(-a) = a, \text{ y}$$

$$a \cdot (-1) \cdot (-1) = (-a) \cdot (-1) = -(-a) = a.$$

Es decir que $(-1) \cdot (-1) = 1$ por unicidad del neutro multiplicativo.

7. Tomando $c = 0$ de la definición de divisibilidad.

8. Sea $a \in R$ con inverso multiplicativo. Sean c y b tales que $a \cdot c = c \cdot a = 1$ y $a \cdot b = b \cdot a = 1$.

$$b = b \cdot (a \cdot c) = (b \cdot a) \cdot c = c.$$

■

Recordemos que un entero es llamado *primo* si no tiene factores no triviales. En general, si R es un anillo, y si $u \in R$ es un elemento que tiene inverso multiplicativo $u^{-1} \in R$, podemos factorizar cualquier elemento $a \in R$ escribiendo $a = u^{-1} \cdot (u \cdot a)$. Los elementos que tienen inverso multiplicativo y los elementos que sólo tienen factorizaciones triviales, tienen un nombre especial.

Definición 2.2.3. Sea R un anillo. Un elemento $u \in R$ es llamado *unidad* si tiene un inverso multiplicativo. Un elemento $a \in R$ se dice *irreducible* si a no es unidad y en toda factorización de a, $a = b \cdot c$, o b es unidad, o c es unidad.

Definición 2.2.4. Sea R un anillo. Y escojamos un elemento no nulo $m \in R$. Decimos que dos elementos a y b de R son *congruentes módulo* m si su diferencia $a - b$ es divisible por m. Escribimos

$$a \equiv b \ (mod\ m)$$

para indicar que a y b son congruentes módulo m.

Las congruencias para anillos arbitrarios satisfacen las mismas propiedades (ecuaciones) que satisfacen en el anillo de los enteros.

Proposición 2.2.5. *Sea R un anillo, y $m \in R$ con $m \neq 0$. Si*

$$a_1 \equiv a_2 \ (mod\ m) \quad y \quad b_1 \equiv b_2 \ (mod\ m)$$

luego

$$a_1 \pm b_1 \equiv a_2 \pm b_2 \ (mod\ m) \quad y \quad a_1 \cdot b_1 \equiv a_2 \cdot b_2 \ (mod\ m).$$

Demostración. Las hipótesis nos dicen que existen $k, h \in R$ tales que

$$a_1 - a_2 = k \cdot m \quad \text{y} \quad b_1 - b_2 = h \cdot m.$$

Entonces, en el primer caso, tenemos que

$$(a_1 + b_1) - (a_2 + b_2) = (a_1 - a_2) + (b_1 - b_2) = k \cdot m + h \cdot m = (k + h) \cdot m.$$

En el segundo caso,

$$(a_1 - b_1) - (a_2 - b_2) = (a_1 - a_2) + (b_2 - b_1) = k \cdot m + (-h) \cdot m = (k - h) \cdot m.$$

Y en el tercero,

$$\begin{gathered}(a_1 \cdot b_1) - (a_2 \cdot b_2) = (k \cdot m + a_2) \cdot b_1 - a_2 \cdot b_2 = k \cdot m \cdot b_1 + a_2 \cdot b_1 - a_2 \cdot b_2 = \\ k \cdot m \cdot b_1 + a_2 \cdot (b_1 - b_2) = k \cdot m \cdot b_1 + a_2 \cdot h \cdot m = m \cdot (k \cdot b_1 + a_2 \cdot h).\end{gathered}$$

■

Teorema 2.2.6. (Teorema chino del resto). Sean $m_1, m_2, \ldots, m_k$ una colección de enteros coprimos dos a dos. Sean $a_1, a_2, \ldots, a_k$ enteros arbitrarios. Entonces el sistema de congruencias simultáneas

$$x \equiv a_1 \ (mod\ m_1),\ \ x \equiv a_2 \ (mod\ m_2), \cdots , x \equiv a_k \ (mod\ m_k).$$

tiene una solución $x = c$. Más aún, si $x = c$ y $x = c\prime$ son ambas soluciones, entonces

$$c \equiv c\prime \ (mod\ m_1 m_2 \dots m_k).$$

Demostración. Supongamos que para algún valor de i hemos podido encontrar una solución $x = c_i$ a las primeras i congruencias simultáneas,

$$x \equiv a_1 \ (mod\ m_1),\ \ x \equiv a_2 \ (mod\ m_2), \cdots , x \equiv a_i \ (mod\ m_i).$$

Por ejemplo, si $i = 1$, entonces $c_1 = a_1$ funciona. Vamos a explicar cómo encontrar solución a una congruencia más:

$$x \equiv a_1 \ (mod\ m_1),\ \ x \equiv a_2 \ (mod\ m_2), \cdots , x \equiv a_{i+1} \ (mod\ m_{i+1}).$$

La idea es encontrar una solución de la forma $x = c_i + m_1 m_2 \dots m_i y$. Notemos que este valor de x aún satisface las primeras i congruencias. Entonces sólo tenemos que elegir y para que también se satisfaga que $x \equiv a_{i+1} \ (mod\ m_{i+1})$. Como sabemos que un entero a tiene inverso multiplicativo módulo m si y sólo si $MCD(a, m) = 1$, y que $MCD(m_{i+1}, m_1 m_2 \cdots m_i) = 1$, entonces siempre podemos hacer esto. Esto completa la prueba de existencia. ■

Nuestra definición de congruencia captura todas las propiedades que vamos a necesitar. Sin embargo, debemos observar que existe una noción más general de congruencia módulo ideales. Para nuestro propósito, es suficiente trabajar con congruencias módulo ideales principales, que son ideales generados por un solo elemento. Una importante consecuencia de la proposición anterior es un método para crear anillos nuevos de anillos viejos, así como creamos $\mathbb{Z}_q$ de $\mathbb{Z}$ mirando las congruencias módulo q.

Definición 2.2.7. Sea R un anillo y $m \in R$ con $m \neq 0$. Para cualquier $a \in R$, escribimos $\overline{a}$ al conjunto de todos los $a\prime \in R$ tales que $a\prime \equiv a \ (mod\ m)$ El conjunto $\overline{a}$ es llamado la *clase de congruencia de* a, y denotamos la colección de todas las clases de congruencias por $R/(m)$ o R/mR. Luego

$$R/(m) = R/mR = \{\overline{a} : a \in R\}.$$

Sumamos y multiplicamos clases de congruencias usando las reglas obvias

$$\overline{a} \oplus \overline{b} = \overline{a + b} \quad \text{y} \quad \overline{a} \odot \overline{b} = \overline{a \cdot b}.$$

Llamamos $R/(m)$ el *anillo cociente de* R *por* m. Este nombre es justificado por la siguiente proposición.

Proposición 2.2.8. *Las fórmulas anteriores inducen una suma y una multiplicación bien definida en el conjunto de todas las clases de congruencia de* $R/(m)$*, y hacen de* $R/(m)$ *un anillo.*

Demostración. Primero probaremos que $\oplus$ y $\odot$ están bien definidas, es decir que dados $\overline{a}$, $\overline{b} \in R/(m)$ y $a\prime \in \overline{a}$, $b\prime \in \overline{b}$, vale que $\overline{a\prime} \oplus \overline{b\prime} = \overline{a} \oplus \overline{b}$ y $\overline{a\prime} \odot \overline{b\prime} = \overline{a} \odot \overline{b}$:

Sean $\overline{a}$, $\overline{b} \in R/(m)$ y $a\prime \in \overline{a}$, $b\prime \in \overline{b}$.

En primer lugar tenemos que $a\prime \equiv a$ y $b\prime \equiv b$ $(mod\ m)$, es decir que existen c_1 y c_2 tales que $a\prime - a = m \cdot c_1$ y $b\prime - b = m \cdot c_2$, o sea $a\prime = m \cdot c_1 + a$ y $b\prime = m \cdot c_2 + b$.

Entonces,

$$\overline{a\prime} \oplus \overline{b\prime} = \overline{a\prime + b\prime} = \overline{a + m \cdot c_1 + b + m \cdot c_2} = \overline{a + b + m \cdot (c_1 + c_2)} = \overline{a + b} \oplus \overline{m \cdot (c_1 + c_2)} = \overline{a + b} \oplus (\overline{m} \odot \overline{c_1 + c_2}) = \overline{a + b} \oplus (\overline{0} \odot \overline{c_1 + c_2}) = \overline{a + b}.$$

Además,

$$\overline{a\prime} \odot \overline{b\prime} = \overline{a\prime \cdot b\prime} = \overline{(a + m \cdot c_1) \cdot (b + m \cdot c_2)} = \overline{a \cdot b + m \cdot c_1 \cdot b + m \cdot c_2 \cdot a + m^2 \cdot c_1 \cdot c_2} = \overline{a \cdot b} \oplus \overline{m \cdot c_1 \cdot b} \oplus \overline{m \cdot c_2 \cdot a} \oplus \overline{m^2 \cdot c_1 \cdot c_2} = \overline{a \cdot b} \oplus (\overline{m} \odot \overline{c_1 \cdot b}) \oplus (\overline{m} \odot \overline{c_2 \cdot a}) \oplus (\overline{m^2} \odot \overline{c_1 \cdot c_2}) = \overline{a \cdot b} \oplus (\overline{0} \odot \overline{c_1 \cdot b}) \oplus (\overline{0} \odot \overline{c_2 \cdot a}) \oplus (\overline{0} \odot \overline{c_1 \cdot c_2}) = \overline{a \cdot b}.$$

Ahora veremos que se cumplen los axiomas de anillo:

- Dados $\overline{a}$, $\overline{b}$, $\overline{c} \in R/(m)$. Veamos que $\oplus$ es asociativa:

$$(\overline{a} \oplus \overline{b}) \oplus \overline{c} = \overline{a + b} \oplus \overline{c} = \overline{(a + b) + c} = \overline{a + (b + c)} = \overline{a} \oplus \overline{b + c} = \overline{a} \oplus (\overline{b} \oplus \overline{c}).$$

- Dado $0 \in R$, veamos que $\overline{0}$ es el neutro de la suma $\oplus$:

$$\text{Sea } \overline{a} \in R/(m),\ \ \overline{a} \oplus \overline{0} = \overline{a + 0} = \overline{a} \ \text{ y } \ \overline{0} \oplus \overline{a} = \overline{0 + a} = \overline{a}.$$

- Dado $\overline{a} \in R/(m)$, como $a \in R$, consideremos $-a$ el opuesto de a, y veamos que $\overline{-a} = -\overline{a}$:

$$\overline{a} \oplus \overline{-a} = \overline{a - a} = \overline{0} \ \text{ y } \ \overline{-a} \oplus \overline{a} = \overline{-a + a} = \overline{0}.$$

- Veamos que $\oplus$ es conmutativa:

$$\overline{a} \oplus \overline{b} = \overline{a + b} = \overline{b + a} = \overline{b} \oplus \overline{a}.$$

- Dados $\overline{a}$, $\overline{b}$, $\overline{c} \in R/(m)$. Veamos que $\odot$ es asociativa:

$$(\overline{a} \odot \overline{b}) \odot \overline{c} = \overline{a \cdot b} \odot \overline{c} = \overline{(a \cdot b) \cdot c} = \overline{a \cdot (b \cdot c)} = \overline{a} \odot \overline{b \cdot c} = \overline{a} \odot (\overline{b} \odot \overline{c}).$$

- Dado $1 \in R$, veamos que $\overline{1} \in R/(m)$ es el neutro del producto $\odot$:

$$\text{Sea } \overline{a} \in R/(m),\ \ \overline{a} \odot \overline{1} = \overline{a \cdot 1} = \overline{a} \ \text{ y } \ \overline{1} \odot \overline{a} = \overline{1 \cdot a} = \overline{a}.$$

- Veamos que el producto $\odot$ distribuye en la suma $\oplus$: Dados $\overline{a}$, $\overline{b}$, $\overline{c} \in R/(m)$

$$\overline{a} \odot (\overline{b} \oplus \overline{c}) = \overline{a} \odot \overline{b + c} = \overline{a \cdot (b + c)} = \overline{a \cdot b + a \cdot c} = \overline{a \cdot b} \oplus \overline{a \cdot c} = (\overline{a} \odot \overline{b}) \oplus (\overline{a} \odot \overline{c}),$$

$$(\overline{a} \oplus \overline{b}) \odot \overline{c} = \overline{a + b} \odot \overline{c} = \overline{(a + b) \cdot c} = \overline{a \cdot c + b \cdot c} = \overline{a \cdot c} \oplus \overline{b \cdot c} = (\overline{a} \odot \overline{c}) \oplus (\overline{b} \odot \overline{c}).$$

■

2.2.1. Anillos de polinomios y el algoritmo de Euclides

Si R es un anillo, podemos crear un anillo polinomial con coeficientes tomados de R. Este anillo es denotado por

$$R[x] = \{a_0 + a_1x + a_2x^2 + \cdots + a_nx^n : n \geqslant 0 \text{ y } a_i \in Ri = 0, \cdots n\}.$$

El *grado* de un polinomio no nulo es el exponente de la potencia más grande de x que aparece. Luego, si

$$\mathbf{a}(x) = a_0 + a_1x + a_2x^2 + \cdots + a_nx^n$$

con $a_n \neq 0$, luego $\mathbf{a}(x)$ tiene grado n. Denotamos el grado de $\mathbf{a}$ por $deg(\mathbf{a})$, y llamamos a a_n el coeficiente principal de $\mathbf{a}(x)$. Un polinomio no nulo cuyo coeficiente principal es 1 es llamado *polinomio mónico*. Son especialmente importantes los anillos de polinomios en los que el anillo R es un cuerpo (para la criptografía, el más importante es el caso de cuerpo $\mathbb{F}_p$). Una razón por la que es muy útil tomar R como un cuerpo, es porque virtualmente todas las propiedades que nos interesan de $\mathbb{Z}$ son válidas para el anillo polinomial inducido. Una de dichas propiedades es el Teorema del Algoritmo de la División (TAD) que tiene su correspondiente en $R[x]$. Podemos hacer lo mismo para cualquier anillo polinomial $\mathbb{F}[x]$ siempre que $\mathbb{F}$ sea un cuerpo. Los anillos de este tipo que tienen un algoritmo de "división con resto" son llamados *anillos Euclideanos*.

Proposición 2.2.9. (El anillo $\mathbb{F}[x]$ es Euclideano). Sea $\mathbb{F}$ un cuerpo y sea $\mathbf{a}$ y $\mathbf{b}$ polinomios en $\mathbb{F}[x]$ con $\mathbf{b} \neq 0$. Luego es posible escribir

$$\mathbf{a} = \mathbf{b} \cdot \mathbf{k} + \mathbf{r} \quad \text{con } \mathbf{k} \text{ y } \mathbf{r} \text{ polinomios, y } \mathbf{r} = 0 \text{ o } deg(\mathbf{r}) < deg(\mathbf{b}).$$

Decimos que $\mathbf{a}$ dividido $\mathbf{b}$ tiene *cociente* $\mathbf{k}$ y *resto* $\mathbf{r}$.

Demostración. Empezamos con cualquier valor para $\mathbf{r}$ y $\mathbf{k}$ satisfaciendo

$$\mathbf{a} = \mathbf{b} \cdot \mathbf{k} + \mathbf{r}.$$

(Por ejemplo, podemos empezar con $\mathbf{k} = 0$ y $\mathbf{r} = \mathbf{a}$). Si $deg(\mathbf{r}) < \mathbf{b})$, entonces ya está. Caso contrario, escribimos

$$\mathbf{b} = b_0 + b_1x + \cdots + b_dx^d \quad \text{y} \quad \mathbf{r} = r_0 + r_1x + \cdots + r_ex^e$$

con $b_d \neq 0$, $r_e \neq 0$ y $e \geqslant d$. Reescribimos la ecuación $\mathbf{a} = \mathbf{b} \cdot \mathbf{k} + \mathbf{r}$ como

$$\mathbf{a} = \mathbf{b} \cdot (\mathbf{k} + \frac{r_e}{b_d}x^{e-d}) + (\mathbf{r} - \frac{r_e}{b_d}x^{e-d} \cdot \mathbf{b}) = \mathbf{b} \cdot \mathbf{k\prime} + \mathbf{r\prime}.$$

Notar que hemos cancelado el término de grado más grande de $\mathbf{r}$, luego $deg(\mathbf{r\prime}) < deg(\mathbf{r})$. Si $deg(\mathbf{r}) < deg(\mathbf{b})$, ya está. Caso contrario, repetimos el proceso. Podemos seguir de este modo mientras $deg(\mathbf{r}) \geqslant deg(\mathbf{b})$, y cada vez que apliquemos este proceso, el grado de $\mathbf{r}$ se hace más chico. Así eventualmente llegamos a un $\mathbf{r}$ cuyo grado es estrictamente menor que el grado de $\mathbf{b}$. ■

Ahora podemos definir divisores comunes y máximo común divisor en $\mathbb{F}[x]$

Definición 2.2.10. Un *divisor común* de dos elementos $\mathbf{a}, \mathbf{b} \in \mathbb{F}[x]$ es un elemento $\mathbf{d} \in \mathbb{F}[x]$ que divide tanto a $\mathbf{a}$ como a $\mathbf{b}$. Decimos que $\mathbf{d}$ es un *máximo común divisor* de $\mathbf{a}$ y $\mathbf{b}$ si todo común divisor de $\mathbf{a}$ y $\mathbf{b}$ también divide a $\mathbf{d}$.

Vamos a ver que cada par de elementos de $\mathbb{F}[x]$ tiene un MCD, que es único salvo multiplicar por una constante de $\mathbb{F}$. Vamos a notar $\text{MCD}(\mathbf{a}, \mathbf{b})$ al único polinomio mónico que es máximo común divisor de $\mathbf{a}$ y $\mathbf{b}$. A priori no es claro que cada par de elementos tenga un MCD, de hecho hay muchos anillos en donde el MCD no existe (por ej. $\mathbb{Z}[x]$). Pero el MCD existe en $\mathbb{F}[x]$ cuando $\mathbb{F}$ es un cuerpo.

Proposición 2.2.11. *(Extensión del algoritmo de Euclides para $\mathbb{F}[x]$) Sea $\mathbb{F}$ un cuerpo y sean $\mathbf{a}$ y $\mathbf{b}$ polinomios en $\mathbb{F}[x]$ con $\mathbf{b} \neq 0$. Luego el MCD($\mathbf{a}, \mathbf{b}$) existe, y existen polinomios $\mathbf{u}$ y $\mathbf{v}$ en $\mathbb{F}[x]$ tales que:*

$$\mathbf{a} \cdot \mathbf{u} + \mathbf{b} \cdot \mathbf{v} = \mathbf{d}.$$

Demostración. El polinomio $\text{MCD}(\mathbf{a}, \mathbf{b})$ se puede calcular aplicando el proceso que se describió en la proposición anterior tal y como se demuestra con los enteros. Simlarmente, los polinomios $\mathbf{u}$ y $\mathbf{v}$ se obtienen sustituyendo una ecuación en la otra como en la demostración del caso entero. ■

Recordemos que un elemento u de un anillo es una unidad si tiene un inverso multiplicativo, y que un elemento a es irreducible si en cualquier factorización $a = bc$ b es unidad o c es unidad. Usando que el grado del producto de dos polinomios es la suma de los grados, se ve que los únicos polinomios que son unidades en $\mathbb{F}[x]$ son las constantes. En cuanto a la irreduciblidad sucede lo mismo que en $\mathbb{Z}$, y la clave para probarlo es el algoritmo extendido de Euclides:

Proposición 2.2.12. *Sea $\mathbb{F}$ un cuerpo. Luego todo polinomio no nulo en $\mathbb{F}[x]$ puede ser factorizado en forma única como producto de polinomios mónicos irreducibles en el siguiente sentido: Si $\mathbf{a} \in \mathbb{F}[x]$ se factoriza como*

$$\mathbf{a} = \alpha \mathbf{p}_1 \cdot \mathbf{p}_2 \cdots \mathbf{p}_m \quad y \quad \mathbf{a} = \beta \mathbf{q}_1 \cdot \mathbf{q}_2 \cdots \mathbf{q}_n$$

donde $\alpha, \beta \in \mathbb{F}$ son constantes y $\mathbf{p}_1, \cdots, \mathbf{p}_m, \mathbf{q}_1, \cdots, \mathbf{q}_n$ son polinomios mónicos irreducibles, entonces luego de un reordenamiento de $\mathbf{q}_1, \cdots, \mathbf{q}_n$, tenemos que

$$\alpha = \beta, \quad m = n, \quad y \quad \mathbf{p}_i = \mathbf{q}_i \text{ para todo } 1 \leq i \leq m.$$

Demostración. La existencia de la factorización se obtiene haciendo inducción en el grado del polinomio y del hecho de que el grado del producto de dos polimios es la suma de los grados. La unicidad se muestra igual que en el caso de los enteros usando el algoritmo extendido de Euclides. ■

2.2.2. Cocientes de anillos de polinomios y cuerpos finitos de orden potencia de primo

En las secciones anteriores, hemos estudiado los anillos polinomiales y los anillos cocientes. Ahora combinaremos estas dos construcciones y consideraremos los cocientes de anillos polinomiales. Recordemos que así como en los enteros módulo m es conveniente trabajar representando cada clase de congruencia módulo m por un entero entre 0 y $m-1$, ahora el algoritmo de división con resto nos va a permitir hacer algo similar para el cociente de anillos polinomiales.

Proposición 2.2.13. *Sea F un cuerpo, y sea $\boldsymbol{m} \in F[x]$ un polinomio no nulo. Luego toda clase de congruencia no nula $\overline{\boldsymbol{a}} \in F[x]/(\boldsymbol{m})$ tiene un único representante $\boldsymbol{r}$ satisfaciendo*

$$deg(\boldsymbol{r}) < deg(\boldsymbol{m}) \quad y \quad \boldsymbol{a} \equiv \boldsymbol{r}\ (mod\ \boldsymbol{m}).$$

Demostración. Usamos que $F[x]$ es Euclideano, como vimos antes, para encontrar polinomios $\mathbf{k}$ y $\mathbf{r}$ tales que

$$\mathbf{a} = \mathbf{m} \cdot \mathbf{k} + \mathbf{r}$$

con $\mathbf{r} = 0$ o $deg(\mathbf{r}) < deg(\mathbf{m})$. Si $\mathbf{r} = 0$, entonces $\mathbf{a} \equiv 0\ (mod\ \mathbf{m})$, luego $\overline{\mathbf{a}} = 0$. En otro caso, reduciendo módulo $\mathbf{m}$ tenemos que $\mathbf{a} \equiv \mathbf{r}\ (mod\ \mathbf{m})$, con el grado de $\mathbf{r}$ menor que el grado de $\mathbf{m}$. Esto muestra que $\mathbf{r}$ existe. Veamos la unicidad: Supongamos que $\mathbf{r'}$ tiene las mismas propiedades. Luego,

$$\mathbf{r} - \mathbf{r'} \equiv \mathbf{a} - \mathbf{a} \equiv 0\ (mod\ \mathbf{m}),$$

luego $\mathbf{m}$ divide a $\mathbf{r} - \mathbf{r'}$. Pero $\mathbf{r} - \mathbf{r'}$ tiene grado estrictamente menor que el grado de $\mathbf{m}$, luego tenemos que $\mathbf{r} - \mathbf{r'} = 0$. ■

Ejemplo 2.2.14. Consideremos el anillo $\mathbb{F}[x]/(x^2+1)$. La proposición anterior dice que cada elemento de este anillo cociente está representado de forma única por un polinomio de la forma

$$\overline{\alpha + \beta x} \quad \text{con} \quad \alpha,\, \beta \in \mathbb{F}.$$

La suma se hace de la forma obvia,

$$\overline{\alpha_1 + \beta_1 x} + \overline{\alpha_2 + \beta_2 x} = \overline{(\alpha_1 + \alpha_2) + (\beta_1 + \beta_2)x}.$$

La multiplicación es similar, excepto que tenemos que dividir el resultado final por x^2+1 y tomar el resto. Así,

$$\overline{\alpha_1 + \beta_1 x} \cdot \overline{\alpha_2 + \beta_2 x} = \overline{\alpha_1\alpha_2 + (\alpha_1\beta_2 + \alpha_2\beta_1)x + \beta_1\beta_2 x^2} = \overline{(\alpha_1\alpha_2 - \beta_1\beta_2) + (\alpha_1\beta_2 + \alpha_2\beta_1)x}.$$

Notemos que el efecto de dividir por x^2+1 es el mismo que reemplazar x^2 por -1. La intuición es que en el anillo cociente $\mathbb{F}[x]/(x^2+1)$, hemos hecho la cantidad x^2+1 igual a 0. Notemos que si tomamos $\mathbb{F} = \mathbb{R}$ en el ejemplo, luego $\mathbb{R}[x]/(x^2+1)$ es simplemente el cuerpo de los números complejos $\mathbb{C}$.

Podemos usar la proposición anterior para contar el número de elementos de un cociente de anillo polinomial cuando $\mathbb{F}$ es un cuerpo finito.

Corolario 2.2.15. *Sea $\mathbb{F}_p$ un cuerpo finito y sea $\boldsymbol{m} \in \mathbb{F}_p[x]$ un polinomio no nulo de grado $d \geqslant 1$. Entonces el anillo cociente $\mathbb{F}_p[x]/(\boldsymbol{m})$ contiene exactamente p^d elementos.*

Demostración. De la proposición anterior sabemos que todo elemento de $\mathbb{F}_p[x]/(\mathbf{m})$ está representado de forma única por un polinomio de la forma

$$a_0 + a_1 x + a_2 x^2 + \cdots + a_{d-1} x^{d-1} \quad \text{con} \quad a_0,\, a_1,\, \cdots a_{d-1} \in \mathbb{F}_p.$$

Hay p posibilidades para a_0, p posibilidades para a_1, y así, lo que nos lleva a un total de p^d elecciones para $a_0,\, a_1,\, \cdots a_{d-1}$. ■

Daremos una importante caracterización de las unidades en un cociente de anillo polinomial, lo que nos permitirá construir nuevos cuerpos finitos.

Proposición 2.2.16. *Sea $\mathbb{F}$ un cuerpo y sean $\boldsymbol{a}$, $\boldsymbol{m} \in \mathbb{F}[x]$ polinomios con $\boldsymbol{m} \neq 0$. Luego, $\overline{\boldsymbol{a}}$ es una unidad en el anillo cociente $\mathbb{F}[x]/(\boldsymbol{m})$ si y sólo si*

$$MCD(\boldsymbol{a}, \boldsymbol{m}) = 1.$$

Demostración. Veamos la doble implicación:

$\Rightarrow$) Supongamos que $\overline{\mathbf{a}}$ es una unidad de $\mathbb{F}[x]/(\mathbf{m})$. Por definición, esto significa que podemos encontrar un $\overline{\mathbf{b}} \in \mathbb{F}[x]/(\mathbf{m})$ satisfaciendo que $\overline{\mathbf{a}} \cdot \overline{\mathbf{b}} = \overline{1}$. En términos de congruencias, esto significa que $\mathbf{a} \cdot \mathbf{b} \equiv 1 \ (mod\ \mathbf{m})$, luego, existe un $\mathbf{c} \in \mathbb{F}[x]$ tal que

$$\mathbf{a} \cdot \mathbf{b} - 1 = \mathbf{c} \cdot \mathbf{m}.$$

Se sigue que cualquier divisor común de $\mathbf{a}$ y de $\mathbf{m}$ también divide a 1. Luego, $MCD(\mathbf{a}, \mathbf{m}) = 1$.

$\Leftarrow$) Ahora supongamos que $MCD(\mathbf{a}, \mathbf{m}) = 1$. Ya vimos que existen polinomios $\mathbf{u}$, $\mathbf{v} \in \mathbb{F}[x]$ tales que

$$\mathbf{a} \cdot \mathbf{u} + \mathbf{m} \cdot \mathbf{v} = 1.$$

Reduciendo módulo $\mathbf{m}$ resulta

$$\mathbf{a} \cdot \mathbf{u} \equiv 1 \ (mod\ \mathbf{m}),$$

luego, $\overline{\mathbf{u}}$ es un inverso para $\overline{\mathbf{a}} \in \mathbb{F}[x]/(\mathbf{m})$. ■

Un caso importante de la proposición anterior es el caso en que el módulo sea un polinomio irreducible.

Corolario 2.2.17. *Sea $\mathbb{F}$ un cuerpo, y sea $\boldsymbol{m} \in \mathbb{F}[x]$ un polinomio irreducible. Luego, el anillo cociente $\mathbb{F}[x]/(\boldsymbol{m})$ es un cuerpo, i.e., todo elemento no nulo de $\mathbb{F}[x]/(\boldsymbol{m})$ tiene un inverso multiplicativo.*

Demostración. Reemplazando $\mathbf{m}$ por un múltiplo constante, podemos asumir que $\mathbf{m}$ es un polinomio mónico. Sea $\overline{\mathbf{a}} \in \mathbb{F}[x]/(\mathbf{m})$. Hay dos casos a considerar. Primero, supongamos que $MCD(\mathbf{a}, \mathbf{m}) = 1$. Luego, la proposición anterior nos dice que $\overline{\mathbf{a}}$ es una unidad, y ya estamos. Segundo, supongamos que $\mathbf{d} = MCD(\mathbf{a}, \mathbf{m}) \neq 1$. Luego, en particular, sabemos que $\mathbf{d} | \mathbf{m}$. Pero $\mathbf{m}$ es mónico e irreducible, y $\mathbf{d} \neq 1$, así que tenemos que $\mathbf{d} = \mathbf{m}$. También sabemos que $\mathbf{d} | \mathbf{a}$, luego $\mathbf{m} | \mathbf{a}$. Así, $\overline{\mathbf{a}} = 0$ en $\mathbb{F}[x]/(\mathbf{m})$. Esto completa la prueba de que todo elemento no nulo de $\mathbb{F}[x]/(\mathbf{m})$ tiene inverso multiplicativo. ■

Si aplicamos este corolario al anillo de polinomios con coeficientes en un cuerpo finito $\mathbb{F}_p$, podemos crear nuevos cuerpos con una cantidad de elementos que es potencia de un primo.

Corolario 2.2.18. *Sea $\mathbb{F}_p$ un cuerpo finito y sea $\boldsymbol{m} \in \mathbb{F}_p[x]$ un polinomio irreducible de grado $d \geqslant 1$. Luego, $\mathbb{F}_p[x]/(\boldsymbol{m})$ es un cuerpo con p^d elementos.*

Demostración. Por el corolario anterior, $\mathbb{F}_p[x]/(\mathbf{m})$ es un cuerpo, y por el corolario anterior a ese, $\mathbb{F}_p[x]/(\mathbf{m})$ tiene p^d elementos. ■

Ejemplo 2.2.19. Veamos que $x^3 + x + 1$ es irreducible en $\mathbb{F}_2[x]$: para eso supongamos que es reducible, y se factoriza como

$$x^3 + x + 1 = (x + \alpha_1)(x + \alpha_2)(x + \alpha_3) = x^3 + x^2(\alpha_1 + \alpha_2 + \alpha_3) + x(\alpha_1 + \alpha_2)\alpha_3 + \alpha_1\alpha_2\alpha_3.$$

Luego, por igualdad de polinomios tenemos el siguente sistema en $\mathbb{F}_2$:

$$\begin{cases} \alpha_1 + \alpha_2 + \alpha_3 &= 0 \\ (\alpha_1 + \alpha_2)\alpha_3 &= 1 \\ \alpha_1\alpha_2\alpha_3 &= 1 \end{cases}$$

Por la segunda ecuación, tenemos que $\alpha_3 = 0$ y que $\alpha_1 + \alpha_2 = 0$, lo que implica que o bien $\alpha_1 = 0$ o $\alpha_2 = 0$. Pero eso contradice la tercera ecuación que dice que $\alpha_1\alpha_2\alpha_3 = 0$. Este absurdo muestra que x^3+x+1 es irreducible en $\mathbb{F}_2[x]$. Luego, $\mathbb{F}_2[x]/(x^3+x+1)$ es un cuerpo con 8 elementos, cuyos representantes son:

$$0,\ 1,\ x,\ x^2,\ 1 + x,\ 1 + x^2,\ x + x^2,\ 1 + x + x^2.$$

La suma y la multiplicación son como las definimos recordando tomar los coeficientes módulo 2.

Ejemplo 2.2.20. Nos preguntamos cuándo el polinomio $x^2 + 1$ es irreducible en $\mathbb{F}_p[x]$. Si es reducible, entonces se factoriza como

$$x^2 + 1 = (x + \alpha)(x + \beta) \quad \text{para } \alpha \text{ y } \beta \in \mathbb{F}_p.$$

Comparando coeficientes, tenemos que $\alpha + \beta = 0$ y $\alpha\beta = 1$; así

$$\alpha^2 = \alpha \cdot (-\beta) = -\alpha\beta = -1.$$

En otras palabras, el cuerpo $\mathbb{F}_p$ tiene un elemento cuyo cuadrado es -1. Recíprocamente, si $\alpha \in \mathbb{F}_p$ satisface que $\alpha^2 = -1$, luego $x^2 + 1 = (x - \alpha)(x + \alpha)$ se factoriza en $\mathbb{F}_p[x]$. Esto prueba que

$$x^2 + 1 \text{ es irreducible en } \mathbb{F}_p[x] \qquad \text{si y sólo si} \qquad -1 \text{ no es un cuadrado en } \mathbb{F}_p.$$

2.3. Cuerpos finitos

Y ahora arrancamos con la artilleria de teoría de cuerpos finitos (que sólo usaremos para demostrar dos teoremas...):

Definición 2.3.1. (Definición 1.1) Decimos que un cuerpo F es una extensión del cuerpo K, siempre que K sea un subcuerpo de F.

Definición 2.3.2. Sea F una extensión del cuerpo K, y $U \subseteq F$. El subcuerpo generado por $K \cap U$ se llama *subcuerpo generado por X sobre K* y se nota $K(X)$. Respectivamente, el subanillo generado por $K \cap U$ se llama *subanillo generado por X sobre K* y se nota $K[X]$.

Definición 2.3.3. (Definición 1.4) Sea F una extensión del cuerpo K, y $u \in F$. Decimos que u es *algebraico sobre K* si es raíz de algún polinomio no nulo $f \in K[x]$.

Teorema 2.3.4. *(Teorema 1.6) Sea F una extensión del cuerpo K y $u \in F$ algebraico sobre K, luego:*

a) $K(u) = K[u]$.

b) $K(u) \cong K[x]/(f)$ *donde* $f \in K[x]$ *es un polinomio irreducible de grado* $n \leqslant 1$ *unicamente determinado por las condiciones* $f(u) = 0$ *y* $g(u) = 0$ *si y solo si* f *divide a* g.

c) $[K(u) : K] = n$.

d) $\{1_K, u, u^2, \cdots, u^{n-1}\}$ *es una base del espacio vectorial* $K(u) sobre K$.

Definición 2.3.5. (Definición 1.7) Sea F una extensión del cuerpo K y $u \in F$ algebraico sobre K. El polinomio mónico irreducible f del teorema 1.6 es llamado el polinomio minimal de u. El grado de u sobre K es $grf = [K(u) : K]$.

Definición 2.3.6. (Definición 3.1) Sea K un cuerpo y $f \in K[x]$, con grado positivo. Una extensión F del cuerpo K se llama *cuerpo de descomposición (splitting field) sobre K del polinomio f* si f se factoriza en $F[x]$ y $F = K(u_1, ..., u_n)$ donde $u_1, ..., u_n$ son las raíces de f en F.

Teorema 2.3.7. *(Corolario 3.9) Sea K un cuerpo y S un conjunto de polinomios con grado positivo en $K[x]$. Luego dos cuerpos de descomposición (splitting fields) de S sobre K son K- isomorfos.*

Teorema 2.3.8. *(Corolario 5.2) Si F es un cuerpo finito entonces $carF = p \neq 0$ para algún p primo y $|F| = p^n$ para algún entero $n \geq 1$. (Donde $carF$ es la caracteristica de F).*

Teorema 2.3.9. *(Corolario 5.6) Sea p un primo y $n \geq 1$ entero. Luego F es un cuerpo finito con p^n elementos si y solo si es el cuerpo de descomposición de $x^{p^n} - x$ sobre $\mathbb{Z}_p$.*

Teorema 2.3.10. *(Corolario 5.8) Si K es un cuerpo finito y $n \geq 1$ un entero, entonces existe una extensión simple $F = K(u)$ de K tal que F es finito y $[F : K] = n$.*

Resumiremos algunas de las propiedades principales de los cuerpos finitos en los próximos dos teoremas:

Teorema 2.3.11. *Sea $\mathbb{F}_p$ un cuerpo finito.*

(a) Para todo $d \geqslant 1$ existe un polinomio irreducible $\boldsymbol{m} \in \mathbb{F}_p[x]$ de grado d.

(b) Para todo $d \geqslant 1$ existe un cuerpo finito con p^d elementos.

(c) Si $\mathbb{F}$ y $\mathbb{F}'$ son cuerpos finitos con el mismo número de elementos, entonces son isomorfos.

Demostración. Por lo visto antes, (a) implica (b). Veamos (a) y (c). Necesitaremos de la artillería que presentamos en la introducción.

Para el (a) sea K cuerpo finito y $d \geq 1$. Por el corolario 5.8 existe una extensión simple del cuerpo, $F = K[u]$, que es finita y con grado de la extensión $[F : K] = d$. Usando el teorema 1.6 y la definición 1.7 , el minimal de u sobre K tiene grado $[F : K] = d$ y es irreducible.

Para demostrar (c) usaremos que todo cuerpo finito F tiene orden p^n con p primo y $n \geq 1$ (Corolario 5.2). Por la proposición 5.6 F es el cuerpo de descomposición de $x^{p^n} - x$ sobre $\mathbb{Z}_p$. Sean F y F' dos cuerpos de orden p^n, entonces son cuerpos de descomposición de $x^{p^n} - x$ sobre $\mathbb{Z}_p$ así por el corolario 3.9 son K isomorfos. ■

Definición 2.3.12. Escribiremos $\mathbb{F}_p^d$ a un cuerpo con p^d elementos. El teorema anterior asegura que existe al menos un cuerpo con p^d elementos, y de haber dos, son esencialmente el mismo. A estos cuerpos se los suele llamar *Cuerpos de Galois* y se los denota $GF(p^d)$.

Observación 2.3.13. Sabemos que si $\mathbb{F}$ es un cuerpo finito, entonces $\mathbb{F}$ tiene p^d elementos, para algún primo p y $d \geq 1$. Y así, el teorema anterior describe todos los cuerpos finitos.

Observación 2.3.14. Para propósitos criptográficos, frecuentemente es ventajoso trabajar en el cuerpo $\mathbb{F}_{2^d}$, que en un cuerpo $\mathbb{F}_p$ con p grande. Esto es debido al hecho de que la naturaleza binaria de las computadoras en general nos permite trabajar con ellas más eficientemente en $\mathbb{F}_{2^d}$. Una segunda razón es que a veces es más útil tener un cuerpo finito que contenga a cuerpos más chicos. En el caso de $\mathbb{F}_{p^d}$, uno puede mostrar que todo cuerpo $\mathbb{F}_{p^e}$ con $e|d$ es un subcuerpo de $\mathbb{F}_{p^d}$. Por supuesto si uno va a usar $\mathbb{F}_{2^d}$ para el intercambio de clave Diffie-Hellman, o el CCP ElGamal, es necesario elegir 2^d aproximadamente del mismo tamaño que uno elegiría p.

Sea $\mathbb{F}$ un cuerpo finito con q elementos. Todo elemento no nulo de $\mathbb{F}$ tiene un inverso, luego el grupo de unidades $\mathbb{F}^*$ es de orden $q - 1$. El teorema de Lagrange nos dice que todo elemento de $\mathbb{F}^*$ tiene un orden que divide a $q - 1$, luego

$$a^{q-1} = 1 \quad \textit{para todo } a \in \mathbb{F}.$$

Esta es una generalización del Pequeño Teorema de Fermat a cuerpos finitos arbitrarios.

Teorema 2.3.15. *Sea $\mathbb{F}$ un cuerpo finito con q elementos. Luego $\mathbb{F}$ tiene una raíz primitiva, i.e., existe un elemento $g \in \mathbb{F}$ tal que*

$$\mathbb{F}^* = \{1, g, g^2, \cdots, g^{q-2}\}.$$

Demostración. Supongamos $q > 2$ (sino es trivial). Sea $h = q - 1$ el orden de $\mathbb{F}^*$, con $h = p_1^{r_1} \cdots p_m^{r_m}$ su descomposición en factores primos.

Para cada $i = 1, ..., m$ $x^{h/p_i} - 1$ tiene a lo sumo h/p_i raíces en $\mathbb{F}$. Como $h/p_i < h$ entonces existen elementos no nulos de $\mathbb{F}$ que no son raíces de este polinomio. Sea a_i uno de ellos.

Definimos $b_i = a_i^{h/p_i^{r_i}}$. Notar que $b_i^{p_i^{r_i}} = 1$, entonces el orden de los b_i divide a $p_i^{r_i}$, es decir, es de la forma $p_i^{s_i}$ con $0 \leq s_i \leq r_i$. Por otro lado $b_i^{p_i^{r_i-1}} = a_i^{h/p_i} \neq 1$. Entonces el orden de b_i es precisamente $p_i^{r_i}$. Llamamos $b = b_1 \cdots b_m$ Veamos que b tiene orden h (ie, va a ser el generador del grupo. Supongamos que no, que el orden de b es un divisor propio de h. Luego es divisor de al menos uno de los m enteros h/p_i. Supongamos sin pérdida de generalidad que es divisor de h/p_1. Luego

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

Si $2 \leq i \leq m$ entonces $p_i^{r_i} | h/p_1$ y así $b_i^{h/p_1} = 1$. Luego $b_1^{h/p_1} = 1$ luego el orden de b_1 divide a h/p_1 Absurdo! pues el orden de b_1 es $p_1^{r_1}$. Así b tiene orden h.

■

2.4. Fórmula de Euler, residuos cuadráticos y reciprocidad cuadrática

Teorema 2.4.1. *(Fórmula de Euler para pq). Sean p y q primos distintos y sea*

$$g = MCD(p-1, q-1).$$

Entonces

$$a^{(p-1)(q-1)/g} \equiv 1 \ (mod\ pq) \quad \textit{para todo a satisfaciendo que } MCD(a, pq) = 1.$$

Demostración. Por hipótesis sabemos que p no divide a a y que g divide a $q-1$, entonces calculamos

$$\begin{array}{rcll} a^{(p-1)(q-1)/g} & = & (a^{(p-1)})^{(q-1)/g} & \text{ya que } (q-1)/g \text{ es entero,} \\ & \equiv & 1^{(q-1)/g} \ (mod\ p) & \text{ya que } a^{(p-1)} \equiv 1 \ (mod\ p) \text{ por el PTF,} \\ & \equiv & 1 \ (mod\ p) & \end{array}$$

El mismo cálculo invirtiendo los roles de p y q muestra que

$$a^{(p-1)(q-1)/g} \equiv 1 \ (mod\ q)$$

Esto prueba que $a^{(p-1)(q-1)/g} - 1$ es divisible tanto por p como por q, y así es divisible por pq; lo que completa la prueba del teorema. ■

Definición 2.4.2. Sea p un primo impar y sea a un número tal que $p \nmid a$. Decimos que a es un *residuo cuadrático módulo p* si a es un cuadrado módulo p, es decir si existe un número c tal que $c^2 \equiv a \ (mod\ p)$.

La siguiente proposición describe qué pasa cuando un residuo cuadrático se multiplica con un número que no lo es.

Proposición 2.4.3. *Sea p un primo impar:*

(a) El producto de dos residuos cuadráticos módulo p es un residuo cuadrático módulo p.

(b) El producto de un residuo cuadrático módulo p y un no residuo cuadrático módulo p no es un residuo cuadrático módulo p.

(c) El producto de dos no residuos cuadráticos módulo p es un residuo cuadrático módulo p.

Demostración. Usaremos un enfoque que nos permitirá demostrar las tres simultáneamente. Sea g una raíz primitiva módulo p, esto es, las potencias $1,\ g,\ g^2,\ \cdots,\ g^{p-2}$ son todas distintas módulo p. ¿Cuáles potencias de g son residuos cuadráticos módulo p? Ciertamente, si $m = 2k$ (es par), luego $g^m = g^{2k} = (g^k)^2$ es un cuadrado. Por otro lado si m es impar, digamos $m = 2k+1$, y supongamos que $g^m \equiv c^2 \ (mod\ p)$, el PTF nos dice que

$$c^{p-1} \equiv 1 \ (mod\ p).$$

Sin embargo, $c^{p-1} \ (mod\ p)$ es también igual a

$$c^{p-1} \equiv (c^2)^{\frac{p-1}{2}} \equiv (g^m)^{\frac{p-1}{2}} \equiv (g^{2k+1})^{\frac{p-1}{2}} \equiv g^{k(p-1)} \cdot g^{\frac{p-1}{2}} \ \ (mod\ p).$$

Otra aplicación del PTF nos dice que

$$g^{k(p-1)} \equiv (g^{p-1})^k \equiv 1^k \equiv 1 \ \ (mod\ p),$$

asi que encontramos que

$$g^{\frac{p-1}{2}} \equiv 1 \ (mod\ p).$$

Esto contradice el hecho de que g es raíz primitiva, lo que prueba que ninguna potencia impar de g es un residuo cuadrático. Hemos probado la siguiente dicotomía: si g es una raíz primitiva módulo p, entonces

$$g^m \begin{cases} \text{es residuo cuadrático} & \text{si } m \text{ es par,} \\ \text{no es residuo cuadrático} & \text{si } m \text{ es impar.} \end{cases}$$

Ahora la proposicion es inmediata, pues en cada caso escribimos a y b como potencias de g, las multiplicamos y sumamos sus exponentes. ■

Esta relación se asemeja a las reglas de multiplicación de 1 *y* -1*, lo que nos lleva a la siguiente definición:*

Definición 2.4.4. Sea p un primo impar, el *símbolo de Legendre* de a es la cantidad $(\frac{a}{p})$ definida por

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

Con esta definición, la proposición anterior se resume en esta simple regla de multiplicación:

$$\left(\frac{a}{p}\right)\left(\frac{b}{p}\right) = \left(\frac{ab}{p}\right)$$

También hacemos la obvia (pero útil) observación de que

$$Si\ a \equiv b\ (mod\ p),\ \ entonces\ \ \left(\frac{a}{p}\right) = \left(\frac{b}{p}\right).$$

Luego, al calcular $\left(\frac{a}{p}\right)$ *reduciremos a a módulo p al intervalo de* 0 *a p. Retornando a nuestra pregunta original de determinar cuándo un número dado es un cuadrado módulo p, el siguiente teorema provee un método para determinar la respuesta.*

Teorema 2.4.5. *(Reciprocidad cuadrática). Sean p y q primos impares.*

(a) $$\left(\frac{-1}{p}\right) = \begin{cases} 1 & si\ p \equiv 1\ (mod\ 4), \\ -1 & si\ p \equiv 3\ (mod\ 4). \end{cases}$$

(b) $$\left(\frac{2}{p}\right) = \begin{cases} 1 & si\ p \equiv 1\ o\ 7\ (mod\ 8), \\ -1 & si\ p \equiv 3\ o\ 5\ (mod\ 8). \end{cases}$$

(c) $$\left(\frac{p}{q}\right) = \begin{cases} \left(\frac{q}{p}\right) & si\ p \equiv 1\ \ o\ q \equiv 1\ (mod\ 4), \\ -\left(\frac{q}{p}\right) & si\ p \equiv 3\ \ y\ q \equiv 3\ (mod\ 4). \end{cases}$$

Demostración. ■

El nombre de reciprocidad cuadrática"proviene de la parte (c), que nos dice cómo se relacionan $(\frac{p}{q})$ *y su recíproco"*$(\frac{q}{p})$*. Indicamos previamente que la reciprocidad cuadrática pùede ser usada para determinar si un número a es un cuadrado módulo p. La forma es aplicar (c) para repetidamente dar vuelta el símbolo de Legendre, donde cada vez que lo damos vuelta, se nos permite reducir el número de arriba modulo el de abajo. Esto conlleva a una rapida reducción en el tamaño de los números. Sin embargo, es posible que durante este proceso sea necesario factorizar números muy grandes, o que nos aparezcan números que no son primos. Así pareciera que la reciprocidad cuadrática es útil sólo si los cálculos intermedios llevan a números que somos capaces de factorizar. Afortunadamente, hay una versión de la reciprocidad cuadrática que elimina completamente esta dificultad. Pero para enunciarla debemos generalizar la definición de símbolo de Legendre.*

Definición 2.4.6. Sean a y b enteros con b impar y positivo. Supongamos que b se factoriza en primos

$$b = p_1^{e_1} p_2^{e_2} \cdots p_t^{e_t}.$$

El *símolo de Jacobi* $(\frac{a}{b})$ esta definido por la fórmula

$$\left(\frac{a}{b}\right) = \left(\frac{a}{p_1}\right)^{e_1} \left(\frac{a}{p_2}\right)^{e_2} \cdots \left(\frac{a}{p_t}\right)^{e_t}.$$

Notar que si b es primo, luego $(\frac{a}{b})$ *es el símbolo de Legendre, así el símbolo de Jacobi es una generalización del de Legendre. Por la definición, pareciera que tenemos que saber factorizar b para calcular el símbolo de Jacobi* $(\frac{a}{b})$*, y que no hemos ganado nada. Sin embargo, veremos que el símbolo de Jacobi tiene más propiedades que el de Legendre, que hacen que los cálculos se vuelvan más rápidos sin ninguna factorización.*

Proposición 2.4.7. *Sean $a,\ a_1,\ a_2,\ b,\ b_1,\ b_2$ enteron con $b,\ b_1,\ b_2$ positivos e impares.*

(a)

$$\left(\frac{a_1a_2}{b}\right)=\left(\frac{a_1}{b}\right)\left(\frac{a_2}{b}\right) \quad y \quad \left(\frac{a}{b_1b_2}\right)=\left(\frac{a}{b_1}\right)\left(\frac{a}{b_2}\right)$$

(b)

$$Si\ a_1=a_2\ (mod\ b),\ entonces\ \left(\frac{a_1}{b}\right)=\left(\frac{a_2}{b}\right).$$

Demostración. Se ve fácilmente por la definición del símbolo de Jacobi y las correspondientes propiedades del símbolo de Legendre. ■

Ahora llegamos al hecho de que el símbolo de Jacobi satisface exactamente la misma reciprocidad que el símbolo de Legendre.

Teorema 2.4.8. *(Reciprocidad cuadrática. Versión II). Sean a y b enteros impares y positivos.*

(a) $$\left(\frac{-1}{b}\right)=\begin{cases}1 & si\ b\equiv 1\ (mod\ 4),\\ -1 & si\ b\equiv 3\ (mod\ 4).\end{cases}$$

(b) $$\left(\frac{2}{b}\right)=\begin{cases}1 & si\ b\equiv 1\ o\ 7\ (mod\ 8),\\ -1 & si\ b\equiv 3\ o\ 5\ (mod\ 8).\end{cases}$$

(c) $$\left(\frac{a}{b}\right)=\begin{cases}\left(\frac{b}{a}\right) & si\ a\equiv 1 \quad o\ b\equiv 1\ (mod\ 4),\\ -\left(\frac{b}{a}\right) & si\ a\equiv 3 \quad y\ b\equiv 3\ (mod\ 4).\end{cases}$$

Demostración. No es difícil usar la versión original de la reciprocidad cuadrática para el símbolo de Legendre para probar esta versión más general para el símbolo de Jacobi. ■

Observación 2.4.9. Supongamos que $\left(\frac{a}{b}\right)=1$, donde b es algún número positivo impar. El hecho de que $\left(\frac{a}{b}\right)=1$, ¿nos dice que a sea un cuadrado módulo b? Lo hace si b es primo, ya que así es como definimos el símbolo de Legendre. Pero, ¿qué pasa si b es compuesto? Por ejemplo, supongamos que $b=pq$ es producto de dos primos. Entonces, por definición,

$$\left(\frac{a}{b}\right)=\left(\frac{a}{pq}\right)=\left(\frac{a}{p}\right)\left(\frac{a}{q}\right).$$

Vemos que hay dos formas en que $\left(\frac{a}{b}\right)$ pueda ser igual a 1, digamos $1=1\cdot 1$ y $1=(-1)\cdot(-1)$. Esto nos lleva a dos casos diferentes:

Caso 1: $\left(\frac{a}{p}\right)=\left(\frac{a}{q}\right)=1$, así que a es un cuadrado módulo pq.

Caso 2: $\left(\frac{a}{p}\right)=\left(\frac{a}{q}\right)=-1$, así que a no es un cuadrado módulo pq.

Deberíamos justificar nuestra aseveración de que a es un cudrado módulo pq en el Caso 1. Notemos que en ese caso, hay soluciones a $c_1^2 \equiv a \ (mod\ p)$ y $c_2^2 \equiv a \ (mod\ q)$. Usamos el TCR para encontrar un entero c satisfaciendo $c \equiv c_1 \ (mod\ p)$ y $c \equiv c_2 \ (mod\ q)$, y luego, $c^2 \equiv a \ (mod\ pq)$. Nuestra conclusiónes que si $b = pq$ es un producto de dos primos, entonces si bien es fácil de computar el valor del símbolo de Jacobi $\left(\frac{a}{pq}\right)$, este valor no nos dice si a es un cuadrado módulo pq. Esta dicotomía puede ser explotada para propósitos criptográficos como se explica en la siguiente sección.

3. Logaritmo Discreto y Diffie-Hellman

3.1. El nacimiento de la criptografía de clave pública

En 1976 Whitfield Diffie y Martin Hellman publiaron un paper titulado "Nuevas Direcciones en Criptografía". Allí, formularon el concepto de un sistema de encriptación de llave pública, y describieron las definiciones y objetivos básicos de un nuevo campo de la matemática y las ciencias de la computación; un campo cuya existencia dependía de la por entonces emergente era de la computadora digital. De todos modos, parece que el concepto de la encriptación de clave pública ya había sido utilizado por miembros del Cuartel General de Comunicaciones del Gobierno Británico, pero al ser información clasificada, esos descubrimientos fueron publicados recién en 1997. La primera contribución importante de Diffie y Hellman en su publicación fue la definición de un Criptosistema de Clve Pública (CCP) y sus componentes asociadas: funciones de un camino e información de atajo. Una *función de un camino* es una función invertible que es fácil de computar pero cuya inversa es difícil de computar. ¿Qué significa que sea difícil de computar? Intuitivamente, una función es difícil de computar si cualquier algoritmo que intente computar la inversa en un tiempo razonable", casi seguramente falle (la frase *casi seguramente* debe ser definida probabilísticamente). Los CCPs seguros se construyen usando funciones de un camino que tienen un atajo, es decir una pieza auxuliar de información que permite que la inversa sea computada fácilmente. De todos modos, hay una enorme diferencia entre la idea abstracta de una función de un camino con atajo y la verdadera construcción de una tal función. Ahora bien, la clave de un CCP consiste de dos partes, una clave privada k_{priv} y una clave pública k_{pub} que se obtiene aplicando algún algoritmo de creación de clave sobre k_{priv}. Para cada par (k_{priv}, k_{pub}) hay un algoritmo de encriptación $e_{k_{pub}}$ y su correspondiente algoritmo de decriptación $d_{k_{priv}}$. El algoritmo de encriptación $e_{k_{pub}}$ correspondiente a k_{pub} es de público conocimiento y fácil de computar. Similarmente, el algritmo de decriptación debe ser fácilmente computable para alguien que conozca la clave privada k_{priv}, pero muy difícil de computar para alguien que solamente conoce k_{pub}. Se dice que la clave privada k_{priv} es *la información atajo* para la función $e_{k_{pub}}$, porque sin esa información de atajo es muy difícil computar la función inversa de $e_{k_{pub}}$, pero con ella se vuelve sencillo. Notemos que en particular, la función que se usa para crear k_{pub} a partir de k_{priv} también debe ser difícil de invertir, ya que k_{pub} es de público conocimiento y k_{priv} es lo que permite una decriptación eficiente. Puede parecer sorprendente, pero aún después de años de investigación todavía no se sabe si las funciones de un camino existen. Se han propuesto varios candidatos de funciones de un camino, y algunas de ellas se utilizan en algoritmos de encriptación modernos. Pero cabe destacar que la seguridad de estos criptosistemas reside en la *asunción* de que invertir la función subyacente es un probema difícil. Diffie y Hellman dieron varias sugerencias de funciones de un camino en su paper, pero no presentaron un ejemplo de CCP principalmente por no encontrar la información de atajo adecuada. De todos modos, describieron un método de clave pública para compartir cierto material de manera segura por medio de un canal inseguro. Su método, llamado ahora *intercambio de clave Diffie-Hellman*, se basa en la idea de que el Problema del Logaritmo Discreto (PLD) es difícil de resolver. Con la publicación de "Nuevas Direcciones en Criptografía.[en] 1976, se inició una carrera por inventar un criptosistema de clave pública eficiente. En tan sólo dos años se publicaron dos papers importantes describiendo CCPs: el

esquema RSA de Rivest, Shamir y Adleman y el esquema "knapsack"de Merkle y Hellman. Pero se descubrió que este último era inseguro en la práctica, mientras que el problema de factorización entera subyacente al RSA es computacionalmente lo suficientemente difícil como para garantizar la seguridad del criptosistema.

3.2. El Problema del Logaritmo Discreto

Definición 3.2.1. Sea $(G, *)$ un grupo, el Problema del Logaritmo Discreto (PLD) para G es determinar, dados g y h elementos en G, un entero x que satisfaga

$$\underbrace{g * g * g * \cdots * g}_{x veces} = h \tag{3.1}$$

La primera publicación acerca de una construcción de clave pública, debida a Diffie y Hellman, está basada en el PLD para el cuerpo finito $\mathbb{F}_p$ (de los enteros modulo p, con p primo).

Teorema 3.2.2. *Sea p un número primo entonces existe un elemento $g \in \mathbb{F}_p^*$ cuyas potencias dan todos los elementos de $\mathbb{F}_p^*$, es decir,*

$$\mathbb{F}_p^* = \{1, g, g^2, g^3, \cdots, g^{p-2}\}. \tag{3.2}$$

Demostración. ■

Aclaración: Los elementos con esta propiedad se llaman raices primitivas o generadores de $\mathbb{F}_p^*$. Y son elementos de orden $p-1$ de $\mathbb{F}_p^*$

Con este teorema vamos a poder definir el PLD para el caso particular del cuerpo $\mathbb{F}_p$

Definición 3.2.3. Sea g una raíz primitiva de $\mathbb{F}_p$ y sea h un elemento no nulo de $\mathbb{F}_p$. El PLD es el problema de encontrar un exponente x tal que

$$g^x \equiv h \ (mod\ p)$$

El número x se llama logaritmo discreto de h en base g y se nota $log_g(h)$.

Teorema 3.2.4. *El logaritmo discreto log_g es un isomorfismo de grupos entre $\mathbb{F}_p^*$ y $\mathbb{Z}_{p-1}$*

Demostración. Notemos que si existe una solución para $g^x \equiv h \ (mod\ p)$, hay infinitas ya que $g^{p-1} \equiv 1 \ (mod\ p)$ por el Pequeño Teorema de Fermat. Entonces si x es solución $x + k(p-1)$ también lo es pues

$$g^{x+k(p-1)} = g^x * (g^{p-1})^k \equiv h * 1^k \equiv h \ (mod\ p)$$

Es decir que $log_g(h)$ está definido módulo p-1. Más aún

$$log_g : \mathbb{F}_p^* \longrightarrow \mathbb{Z}_{p-1}$$

está bien definida. ■

3.3. Intercambio de clave Diffie-Hellman

El intercambio de clave Diffie-Hellman resuelve el siguiente problema: Ana y Bruno quieren compartir una clave simétrica (es decir que se usa la misma clave para encriptar que para desencriptar el mensaje), pero todos sus medios de comunicación son inseguros, o sea que cualquier información que intercambien será observada por su adversario Inés. ¿Cómo es posible para Ana y Bruno compartir una clave sin que Inés la conozca? En principio parece un problema imposible, pero Diffie y Hellman aprovecharon la dificultad del PLD en el cuerpo $\mathbb{F}_p^*$ para dar una posible solución. El primer paso es que Ana y Bruno se pongan de acuerdo (de manera pública) en un primo grande p, y un entero no nulo g, preferentemente tal que $ord(g)$ en $\mathbb{F}_p^*$ sea un primo grande. Lo siguiente que tienen que hacer es cada uno elegir un número que será secreto, digamos que Ana elige a y Bruno b. A continuación cada uno computa lo siguiente

$$\underbrace{A \equiv g^a \ (mod\ p)}_{\text{Ana computa esto}} \quad \text{y} \quad \underbrace{B \equiv g^b \ (mod\ p)}_{\text{Bob computa esto}}$$

Luego intercambian esta información, Ana le envía A a Bruno, y Bruno B a Ana. Notemos que inés también conoce los valores de A y B, dado que el canal de comunicación es inseguro. Finalmente, Ana y Bruno utilizan nuevamente sus claves privadas para computar

$$\underbrace{A' \equiv B^a \ (mod\ p)}_{\text{Ana computa esto}} \quad \text{y} \quad \underbrace{B' \equiv A^b \ (mod\ p)}_{\text{Bob computa esto}}$$

Estos valores que ellos computan, A' y B', son en realidad iguales módulo p, pues

$$A' \equiv B^a \equiv (g^b)^a \equiv g^{ab} \equiv (g^b)^a \equiv B^a \equiv B' \ (mod\ p)$$

Este valor común que intercambiaron será su clave simétrica. En general, el dilema de Inés es el siguiente: ella conoce los valores de A y B, con lo cual conoce los valores de g^a y g^b. También conoce los valores de g y p, así que si pudiera resolver el PLD, podría encontrar a y b respectivamente y usarlos para calcular la clave secreta de Ana y Bruno, g^{ab}. Parecería que la clave de Ana y Bruno está segura mientras que Inés no pueda resolver el PLD, pero esto no es tan así. Es cierto que una forma de encontrar la clave es resolviendo el PLD, pero no es ese precisamente el problema que debe resolver Inés. La seguridad de la clave de Ana y Bruno reside en el siguiente problema (posiblemente más sencillo):

Definición 3.3.1. Sean p primo y g entero. El *Problema Diffie-Hellman* (PDH) es el problema de computar g^{ab} $(mod\ p)$ conociendo los valores de g^a $(mod\ p)$ y g^b $(mod\ p)$.

Está claro que el PDH no es más difícil que el PLD, pues si Inés puede resolver el PLD, puede resolver el PDH. Pero la recíproca no es tan clara: Supongamos que Inés tiene un algoritmo que le permite resolver el PDH, ¿puede con eso resolver eficientemente el PLD? La respuesta no se conoce.

3.4. El criptosistema de clave pública ElGamal

Si bien el algoritmo Diffie-Hellman nos da un método para compartir una clave secreta a través de un medio público, no es estrictamente un CCP. El primer CCP conocido fue el RSA de 1978, pero el CCP descripto por Taher ElGamal en 1985 está más relacionado con el PLD y el Algoritmo de Diffie-Hellman. En esta sección vamos a describir el CCP de ElGamal basado en el PLD sobre el cuerpo $\mathbb{F}_p^*$, pero la construcción puede usarse en general para cualquier grupo. Digamos que Ana y Bruno no se conocen, pero Bruno quiere mandarle un mensaje encriptado a Ana. Para esto Ana publica su clave pública y un algoritmo, luego Bruno encripta su mensaje utilizando la clave y el algoritmo de Ana. Finalmente Ana, con su clave privada puede desencriptar el mensaje que le envió Bruno. Veamos cúales son las claves y los algoritmos en el caso particular del CCP de ElGamal. Ana necesitará un número primo p grande para el cual el PLD sea dificil en $\mathbb{F}_p^*$ y un elemento g módulo p de orden primo y grande. Además elige un número secreto a que será su clave privada, y calcula

$$A \equiv g^a \ (mod\ p)$$

Ana publica su clave pública A. Ahora si Bruno desea encriptar un mensaje utilizando la clave pública de Ana hace lo siguiente: Transforma su mensaje en un número m entre 2 y p (esto es posible a través de distintos esquemas de cifrado). Para encriptar m, primero elige otro número k módulo p y calcula las dos cantidades

$$c_1 \equiv g^k \ (mod\ p) \ \text{ y } \ c_2 \equiv mA^k \ (mod\ p).$$

El mensaje cifrado que le envía Bruno a Ana es el par de números (c_1, c_2).
Para descifrar el mensaje, Ana calcula $x \equiv c_1^a \ (mod\ p)$, y $x^{-1} \ (mod\ p)$. Luego multiplica c_2 por x^{-1} y el resultado es el mensaje m. Para ver esto observemos:

$$\begin{array}{rcccc} x^{-1} \cdot c_2 & \equiv & (c_1^a)^{-1} & (mod\ p), & \text{pues } x \equiv c_1^a \ (mod\ p), \\ & \equiv & (g^{ak})^{-1} \cdot (mA^k) & (mod\ p), & \text{pues } c_1 \equiv g^k, c_2 \equiv mA^k \ (mod\ p), \\ & \equiv & (g^{ak})^{-1} \cdot (m(g^a)^k) & (mod\ p), & \text{pues } A \equiv g^a \ (mod\ p) \\ & \equiv & m & (mod\ p), & \text{pues el término } g^{ak} \text{ se cancela.} \end{array}$$

¿Qué tendría que hacer Inés si quisiera decriptar el mensaje? Inés conoce los parámetros públicos p y g y el valor de $A \equiv g^a \ (mod\ p)$. Si Inés pudiera resolver el PLD podría encontrar a y decriptar el mensaje. De otra manera parece difícil que Inés pueda encontrar una forma de conocer el mensaje m. Veamos ahora la relación que hay entre ElGamal y el problema de Diffie-Hellman:

Proposición 3.4.1. *Fijar un primo p y una base g para usar en la encriptación de ElGamal. Supongamos que Inés tiene acceso a un oráculo que decripta textos cifrados con ElGamal a partir de las claves públicas. Entonces puede usar ese oráculo para resolver el problema de Diffie-Hellman*

Demostración. Recordemos que en el problema de Diffie-Hellman Inés tiene los valores

$$A \equiv g^a \ (mod\ p) \ \text{ y } \ B \equiv g^b \ (mod\ p)$$

(pero no conoce los valores a y b) y quiere calcular el valor de g^{ab} $(mod\ p)$.
Ahora supongamos que Inés puede consultar el oráculo de ElGamal, es decir, le envía al oráculo un primo p, una base g, una llave pública A y un texto cifrado (c_1, c_2) y el oráculo le devuelve el valor $(c_1^a)^{-1} \cdot c_2$ $(mod\ p)$.
Si Inés quisiera resolver el problema de Diffie-Hellman, ¿qué valores de c_1 y c_2 debería elegir? Tomando un c_2 arbitrario le entrega al oráculo el texto cifrado (B, c_2) el oráculo entonces le devuelve el mensaje m que satisface

$$m \equiv\ (c_1^a)^{-1} \cdot c_2 \equiv (B^a)^{-1} \cdot c_2 \equiv (g^{ab})^{-1} \cdot c_2\ (mod\ p).$$

Así Inés alcula $m^{-1} \cdot c_2 \equiv g^{ab}$ $(mod\ p)$ para encontrar el valor de g^{ab} $(mod\ p)$.
Notemos que Inés resolvió el problema de Diffie-Hellman pero no el PLD. ■

3.5. ¿Qué tan díficil es el PLD?

Dado un grupo G y dos elementos g, h en G el PLD busca un exponente x tal que $g^x = h$. ¿Cómo podemos cuantificar la dificultad de este problema? Una medida natural de la dificultad es aproximar el número de operaciones necesarias para resolver el problema usando el método más eficiente conocido. Por ejemplo, supongamos que contamos el proceso de computar g^x como una sola operación, entonces un intento a prueba y error de resolver el PLD sería computar g^x para $x = 1, 2, ..$ y comparar los valores con h. Si g tiene orden n, entonces este algortitmo garantiza encontrar la solución en a lo sumo n operaciones, pero si n es grande, digamos $n > 2^80$, no es un algoritmo práctico con los métodos computacionales de hoy en día.

Definición 3.5.1. (Notación de Orden). Sean $f(x)$ y $g(x)$ funciones de x positivas. Decimos que "f es del orden de g" y escribimos

$$f(x) = \mathcal{O}(g(x))$$

si hay constantes positivas c y C

$$f(x) \leq cg(x) \text{ para todo } x \geq C.$$

En particular, escribimos que $f(x) = \mathcal{O}(1)$ si $f(x)$ está acotada para todo $x \geq C$

La siguiente proposición nos da un método para probar que $f(x) = \mathcal{O}(g(x))$.

Proposición 3.5.2. *Si el límite*

$$\lim_{x \longrightarrow \infty} \frac{f(x)}{g(x)}$$

existe y es finito, entonces $f(x) = \mathcal{O}(g(x))$

Demostración. Sea L el límite. Por definición de límite, para todo $\varepsilon > 0$ existe una constante C_ε tal que

$$|\frac{f(x)}{g(x)} - L| < \varepsilon \text{ para todo } x > C_\varepsilon.$$

En particular, tomando $\varepsilon = 1$, encontramos que

$$\frac{f(x)}{g(x)} < L + 1 \text{ para todo } x > C_1$$

Así, por definición, $f(x) = \mathcal{O}(g(x))$ con $c = L + 1$ y $C = C_1$. ■

La notación de orden nos permite definir varios conceptos relacionados con la complejidad computacional de problemas matemáticos. Supongamos que estamos tratando de resolver un cierto tipo de problema donde la entrada es un número cuyo tamaño puede variar; nos interesa saber cuánto tarda en resolverse el problema en términos del tamaño de la entrada (típicamente se mide el tamaño de la entrada por su cantidad de bits). Supongamos que hay una constante $A \geq 0$, independiente del tamaño de la entrada, tal que si la entrada es de tamaño $\mathcal{O}(k)$, lleva $\mathcal{O}(k^A)$ pasos resolver el problema. Entonces decimos que el problema se resuelve en *tiempo polinómico*. Si podemos tomar $A = 1$, decimos que se resuelve en *tiempo lineal*, si podemos tomar $A = 2$, se resuelve en *tiempo cuadrático*. Los algoritmos de tiempo polinomial son considerados algoritmos rápidos. Por otro lado, si hay una constante $c > 0$ tal que si la entrada es de tamaño $\mathcal{O}(k)$, hay un algoritmo que resuelve el problema en $\mathcal{O}(e^{ck})$ pasos, entonces decimos que el problema se resuelve en *tiempo exponencial*. Los algoritmos de tiempo exponencial son considerados algoritmos lentos. Entre ellos se encuentran los algoritmos de tiempo sub-exponencial. Estos tienen la propiedad de que para todo $\varepsilon > 0$ resuelven el problema en $\mathcal{O}_\varepsilon(e^{k\varepsilon})$ pasos. Esta notación significa que las constantes c y C que aparecen en la definición, pueden depender del ε.

3.6. Un algoritmo de colisión para el PLD

En esta sección describiremos un algoritmo para el logaritmo discreto desarrollado por Shanks. Primero comenzaremos estableciendo el tiempo de ejecución de un algoritmo de fuerza bruta trivial para resolver el PLD.

Proposición 3.6.1. (Cota trivial para el PLD). Sea G un grupo, y $g \in G$ un elemento de orden N. Entonces el problema del logaritmo discreto

$$g^x = h$$

puede resolverse en $\mathcal{O}(N)$ pasos, donde cada paso consiste de una multiplicación por g.

Demostración. Simplemente hacemos una lista de los valores de g^x para $x = 0, 1, 2, \cdots, N-1$. Notemos que cada valor sucesivo puede obtenerse multiplicando por g al anterior. Si existe una solución para $g^x = h$, aparecerá en la lista. ■

Observación 3.6.2. Si trabajamos en $\mathbb{F}_p^*$, cada cómputo de g^x $(mod\ p)$ requiere $\mathcal{O}((log\ p)^k)$ operaciones de la computadora, donde la constante k y la constante en la notación de $\mathcal{O}$ dependen de la computadora y del algoritmo utilizado para la multiplicación modular. En este caso, el numero total de pasos para ejecutar el PLD (tiempo de ejecución) sería de $\mathcal{O}(N(log\ p)^k)$.

La idea de un algoritmo de colisión es hacer dos listas y buscar un elemento que aparezca en ambas. Para el algoritmo descripto anteriormente, el tiempo de ejecución de un algoritmo de colisión sería de un poquito más que $\mathcal{O}(\sqrt{N})$ pasos, que es una enorme ventaja frente a $\mathcal{O}(N)$ si N es grande.

Proposición 3.6.3. (Algoritmo Paso de bebé-Paso de gigante de Shanks). Sea G un grupo, y sea $g \in G$ un elemento de orden $N \geqslant 2$. El siguiente algoritmo resuelve el PLD $g^x = h$ en $\mathcal{O}(\sqrt{N} \cdot log\ N)$ pasos.

1. Sea $n = 1 + \lfloor \sqrt{N} \rfloor$, así que en particular $n > \sqrt{N}$
2. Crear dos listas,

$$\begin{array}{ll} \text{Lista 1:} & e,\ g,\ g^2,\ g^3, \cdots,\ g^n \\ \text{Lista 2:} & h,\ h \cdot g^{-n},\ h \cdot g^{-2n}, \cdots,\ h \cdot g^{-n^2} \end{array}$$

3. Encontrar una coincidencia entre las listas, digamos $g^i = hg^{-jn}$
4. Luego, $x = i + jn$ es una solución para $g^x = h$

Demostración. Empezaremos con algunas observaciones. Primero, cuando creamos la Lista 2, empezamos por computar el valor $u = g^{-n}$ y luego compilamos la Lista 2 computando $h,\ h \cdot u,\ h \cdot u^2,\ \cdots,\ h \cdot u^n$. Así, crear las dos listas lleva aproximadamente $2n$ multiplicaciones (multiplicar por g es un paso de bebé y multiplicar por $u = g^{-n}$ es un paso de gigante, de ahí el nombre). Segundo, si existe una coincidencia, podemos encontrarla en un pequeño múltiplo de $log(n)$ pasos usando algoritmos de búsqueda estándar, así que el paso 3 lleva $\mathcal{O}(log\ n)$ pasos. Así que el tiempo total de ejecución del algoritmo es de $\mathcal{O}(n\ log\ n) = \mathcal{O}(\sqrt{N}\ log\ N)$. Para este último paso usamos el hecho de que $n \approx N$, y entonces

$$n\ log\ n \approx \sqrt{N}\ log\ \sqrt{N} = \frac{1}{2}\sqrt{N}\ log\ N.$$

Para probar que el algoritmo funciona, debemos mostrar que las dos listas siempre tienen una coincidencia. Para ver esto, sea x la solución desconocida para $g^x = h$ y escribamos x como

$$x = nq + r \quad \text{con} \quad 0 \leqslant r < n.$$

Sabemos que $1 \leqslant x < N$, así que

$$q = \frac{x - r}{n} < \frac{N}{n} < n \quad \text{pues} \quad n > \sqrt{N}$$

Entonces podemos reescribir la ecuación $g^x = h$ como

$$g^r = h \cdot g^{-qn} \quad \text{con} \quad 0 \leqslant r < n \ \text{ y } \ 0 \leqslant q < n.$$

Así, g^r está en la Lista 1 y $h \cdot g^{-qn}$ está en la Lista 2, lo que muestra que las Listas 1 y 2 tienen un elemento en común. ■

3.7. El teorema chino del resto

El TCR describe las soluciones a un sistema de congruencias lineales simultáneas. La situación más sencilla es un sistema de dos congruencias

$$x \equiv a \ (mod\ m) \ \text{ y } \ x \equiv b \ (mod\ n)$$

con $MCD(m, n) = 1$, en cuyo caso el TCR dice que la solución es única módulo mn. Comenzaremos con un ejemplo en el que resolveremos dos congruencias simultáneas, que después generalizaremos para resolver el problema del TCR.

Ejemplo 3.7.1. Buscamos un x que resuelva simultáneamente

$$x \equiv a \ (mod\ m) \ \text{ y } \ x \equiv b \ (mod\ n)$$

La primer congruencia dice que como $x \equiv a \ (mod\ m)$, el conjunto de soluciones de la primer congruencia es el de los enteros

$$x = a + my, \ \ y \in \mathbb{Z}.$$

Sustituyendo en la segunda congruencia, tenemos

$$a + my \equiv b \ (mod\ n), \ \text{ y así } \ my \equiv b - a \ (mod\ n).$$

Resolvemos para y, multiplicando a ambos lados por el inverso de m módulo n, que existe porque el $MCD(m, n) = 1$, y que es m^{-1}. Obtenemos

$$y \equiv (b - a) \cdot m^{-1} \ (mod\ n).$$

Finalmente, reemplazando este valor de y en la primera ecuación, tenemos la solución

$$x = a + m \cdot (b - a) \cdot m^{-1}$$

al problema original.

Observación 3.7.2. Notemos que en la solución del ejemplo anterior, m^{-1} es el inverso multiplicativo de $m \ (mod\ n)$. Y es por eso que no se cancelan considerando la ecuación en $\mathbb{Z}$.

En la sección prelimiares hemos citado y demostrado el TCR, y se puede observar que la misma demostración provee un método concreto para hallar la solución como hemos visto en el ejemplo.

3.7.1. Resolviendo congruencias con módulos compuestos

En general es más fácil resolver una congruencia con módulo compuesto primero resolviendo varias congruencias módulo primos (o potencias de primos) y luego uniéndolas usando el TCR. En esta sección ilustraremos el principio discutiendo el problema de encontrar raíces cuadradas módulo m. Resulta que es relativamente sencillo computar las raíces cuadradas módulo un primo. De hecho, para primos congruentes con 3 módulo 4 es extremadamente sencillo encontrar sus raíces cuadradas, como veremos en la siguiente proposición:

Proposición 3.7.3. *Sea p un primo satisfaciendo $p \equiv 3 \pmod 4$. Sea a un entero tal que la congruencia $x^2 \equiv a \pmod p$ tiene solución (es decir tal que a tiene una raiz cuadrada módulo p). Entonces*

$$b \equiv a^{(p+1)/4} \pmod p$$

es una solución; satisface que $b^2 \equiv a \pmod p$.

Notemos que esta proposición es válida sólo si a tiene una raíz cuadrada módulo p. Más adelante daremos un método eficiente para determinar si un número tiene o no una raíz cuadrada módulo p.

Demostración. Sea g una raíz primiva módulo p. Entonces a es alguna potencia de g, y el hecho de que a tenga una raíz cuadrada módulo p implica que a es una potencia par de g, digamos $a \equiv g^{2k} \pmod p$. Calculamos

$$\begin{array}{rcccl} b^2 & \equiv & a^{\frac{p+1}{2}} & (\mathit{mod}\ p) & \text{definición de } b, \\ & \equiv & (g^{2k})^{\frac{p+1}{2}} & (\mathit{mod}\ p) & \text{pues } a \equiv g^{2k} \ (\mathit{mod}\ p), \\ & \equiv & g^{(p+1)k} & (\mathit{mod}\ p) & \\ & \equiv & g^{2k+(p-1)k} & (\mathit{mod}\ p) & \\ & \equiv & a \cdot (g^{p-1})^k & (\mathit{mod}\ p) & \text{pues } a \equiv g^{2k} \ (\mathit{mod}\ p), \\ & \equiv & a & (\mathit{mod}\ p) & \text{pues } g^{p-1} \equiv 1 \ (\mathit{mod}\ p). \end{array}$$

Luego b es una raíz cuadrada de a módulo p. ■

Ejemplo 3.7.4. Buscamos una solución a la congruencia

$$x^2 \equiv 197 \pmod{437}$$

El módulo se factoriza como $437 = 19 \cdot 23$, así que primero resolveremos las dos congruencias

$$y^2 \equiv 197 \equiv 7 \pmod{19} \quad \text{y} \quad z^2 \equiv 197 \equiv 13 \pmod{23}.$$

Como 19 y 23 son ambos congruentes módulo 3 con 4, podemos encontrar sus raíces cuadradas usando la proposición anterior. Teniendo así

$$y \equiv \pm 8 \pmod{19} \quad \text{y} \quad z \equiv \pm 6 \pmod{23}.$$

Podemos tomar 8 o -8 para y, y 6 o -6 para z. Eligiendo las dos soluciones positivas, usamos el TCR para resolver las congruencias simultáneas

$$y \equiv 8 \pmod{19} \quad \text{y} \quad z \equiv 6 \pmod{23}.$$

Encontramos que $x \equiv 236 \pmod{437}$ nos da la solución deseada.

Observación 3.7.5. Es claro por el ejemplo que es relativamente sencillo calcular las raíces cuadradas módulo m en producto de potencias de primos. Sin embargo, si m es muy grande y no podemos factorizarlo, entonces encontrar las raíces cuadradas módulo m es un problema muy difícil. De hecho, si m es un número compuesto muy grande, cuya factorización es desconocida, resulta muy difícil determinar si un entero a tiene una raíz cuadrada módulo m, incluso sin pedir calcularla. El CCP Goldwasser-Micali (que describiremos más adelante) está basado en la dificultad de identificar qué numeros tienen raíz cuadrada módulo un número compuesto m. La información de atajo es el conocimiento de los factores de m.

3.8. El algoritmo de Pohlig-Hellman

Notemos que si $m = m_1 \cdot m_2 \cdots m_t$ es un producto de enteros coprimos dos a dos entonces el TCR, dice que resolver una ecuación módulo m es más o menos equivalente a resolver la ecuación módulo m_i para cada i, ya que el teorema nos dice como unir las soluciones para obtener una solución módulo m. Si G es un grupo y $g \in G$ es un elemento de orden N entonces las soluciones para $g^x = h$ en G están determinadas unicamente módulo N, asi que la factorización prima de N, parece ser relevante. Esta es la idea central del algoritmo de Pohlig-Hellman.

Teorema 3.8.1. (Algoritmo de Pohlig-Hellman). Sea G un grupo y supongamos que tenemos un algoritmo para resolver el PLD en G para un elemento cuyo orden es una potencia de un primo. Concretamente si $g \in G$ tiene orden q^e, supongamos que podemos resolver $g^x = h$ en $\mathcal{O}(S_{q^e})$ pasos. Ahora, sea $g \in G$ un elemnto de orden N y supongamos que N se factoriza como producto de potencias de primos así

$$N = q_1^{e_1} \cdot q_2^{e_2} \cdots q_t^{e_t}.$$

Entonces el PLD $g^x = h$ se puede resolver en

$$\mathcal{O}(\sum_{i=1}^{t} S_{q_i^{e_i}} + log\ N)$$

pasos, usando el siguiente procedimiento:

1. Para cada $1 \leqslant i \leqslant t$, sea

   $$g_i = g^{N/q_i^{e_i}} \quad \text{y} \quad h_i = h^{N/q_i^{e_i}}.$$

   Notemos que g_i tiene orden $q_i^{e_i}$, así que usamos el algoritmo dado para resolver el PLD

   $$g_i^y = h_i.$$

   Sea $y = y_i$ una solución a la ecuación anterior.

2. Usar el TCR para resolver

   $$x \equiv y_1\ (mod\ q_1^{e_1}),\ x \equiv y_2\ (mod\ q_2^{e_2}),\ \cdots,\ x \equiv y_t\ (mod\ q_t^{e_t}).$$

Demostración. La cantidad de pasos es clara, ya que el item (1) toma $\mathcal{O}(\sum S_{q_i^{e_i}})$ pasos y el item (2), usando el TCR, toma $\mathcal{O}(log\ N)$ pasos. Nos queda probar que los items (1) y (2) nos dan una solución a $g^x = h$. Sea x una solución al último sistema de congruencias. Luego, para cada i podemos escribir

$$x = y_i + q_i^{e_i} z_i \text{ para algún } z_i.$$

Esto nos permite computar

$$\begin{aligned}
(g^x)^{N/q_i^{e_i}} &= (g^{y_i+q_i^{e_i}z_i})^{N/q_i^{e_i}} && \text{por lo anterior}\\
&= (g^{N/q_i^{e_i}})^{y_i}\cdot g^{Nz_i} &&\\
&= (g^{N/q_i^{e_i}})^{y_i} && \text{ya que } g^N \text{ es el neutro}\\
&= g_i^{y_i} && \text{por definición de } g_i,\\
&= h_i && \text{por la definición del paso (1)}\\
&= h^{N/q_i^{e_i}} && \text{por definición de } h_i.
\end{aligned}$$

En términos del logaritmo discreto para la base g, podemos reescribir esto como

$$\frac{N}{q_i^{e_i}}\cdot x \equiv \frac{N}{q_i^{e_i}}\cdot log_g(h)\ (mod\ N), \qquad (*)$$

donde recordemos que el logaritmo discreto en base g está definido solamente módulo N, ya que g^N es el neutro. Ahora observemos que los números

$$\frac{N}{q_1^{e_1}},\ \frac{N}{q_2^{e_2}},\ \cdots,\ \frac{N}{q_t^{e_t}}$$

tienen MCD 1. Por lo tanto existen enteros $c_1, c_2, \cdots, c_t$ tales que

$$\frac{N}{q_1^{e_1}}\cdot c_1 + \frac{N}{q_2^{e_2}}\cdot c_2 + \cdots + \frac{N}{q_t^{e_t}}\cdot c_t = 1.$$

Ahora multiplicamos ambos lados de (*) por c_i y sumando sobre $i = 1, 2, \cdots, t$, obtenemos

$$\sum_{i=1}^{t}\frac{N}{q_i^{e_i}}\cdot c_i\cdot x \equiv \sum_{i=1}^{t}\frac{N}{q_i^{e_i}}\cdot c_i\cdot log_g(h)\ (mod\ N)$$

,

y luego, lo anterior nos dice que

$$x = log_g(h)\ (mod\ N).$$

Esto prueba que x satisface que $g^x \equiv h$. ■

Observación 3.8.2. El algoritmo de Pohlig-Hellman reduce de alguna manera el PLD para elementos de orden arbitrario al PLD para elementos de orden de potencias de primos. Un mejor refinamiento esencialmente reduce el problema a elementos de orden primo, más precisamente en la notación del teorema 9.0.4, la cantidad de pasos S_q^e para elementos de orden q^e puede reducirse a $\mathcal{O}(eS_q)$. El algoritmo de Pohlig-Hellman nos dice que el PLD en un grupo G no es seguro si el orden del grupo es un producto de potencias de primos pequeños. Más generalmente, $g^x = h$ es fácil de resolver si el orden del elemento g es un producto de potencias de primos pequeños.

Ahora explicaremos el algoritmo que reduce el PLD para elementos de orden de potencias de un primo al PLD para elementos de orden primo. La idea es simple: Si g tiene orden q^e, entonces $g^{q^{e-1}}$ tiene orden q. El truco es repetir el proceso varias veces y luego acomodar la información en la respuesta final.

Proposición 3.8.3. *Sea G un grupo. Supongamos que q es primo, y que conocemos un algoritmo que lleva S_q pasos para resolver el PLD $g^x = h$ en G siempre que g tenga orden q. Ahora, sea $g \in G$ un elemento de orden q^e con $e \geqslant 1$. Entonces podemos resolver el PLD*

$$g^x = h \quad en \quad \mathcal{O}(eS_q) \quad pasos.$$

Demostración. La idea de la prueba es escribir el exponente x desconocido de la forma

$$x = x_0 + x_1 q + x_2 q^2 + \cdots + x_{e-1} q^{e-1} \quad \text{con} \quad 0 \leqslant x_i < q,$$

y luego determinar sucesivamente $x_0, x_1, x_2, \cdots$. Empezaeremos observando que el elemento $g^{q^{e-1}}$ es de orden q. Lo que nos permite computar

$$\begin{aligned} h^{q^{e-1}} &= (g^x)^{q^{e-1}} && \text{elevando ambos miembros de } g^x = h \\ &= (g^{x_0 + x_1 q + x_2 q^2 + \cdots + x_{e-1} q^{e-1}})^{q^{e-1}} && \text{por como escribimos } x \\ &= g^{x_0 q^{e-1}} \cdot (g^{q^e})^{x_1 + x_2 q + \cdots + x_{e-1} q^{e-2}} \\ &= (g^{q^{e-1}})^{x_0} && \text{ya que } g^{q^e} = 1 \end{aligned}$$

Como $g^{q^{e-1}}$ es un elemento de orden q en G, la ecuación

$$(g^{q^{e-1}})^{x_0} = h^{q^{e-1}}$$

es el PLD cuya base es un elemento de orden q. Por hipótesis, podemos resolver el problema en S_q pasos. Una vez hecho esto, conocemos un exponente x_0 con la propiedad de que

$$g^{x_0 q^{e-1}} = h^{q^{e-1}} \quad \text{en} \quad G$$

Hacemos un cálculo similar, esta vez elevando ambos lados de $g^x = h$ a la q^{e-2} lo que establece

$$\begin{aligned} h^{q^{e-2}} &= (g^x)^{q^{e-2}} \\ &= (g^{x_0 + x_1 q + x_2 q^2 + \cdots + x_{e-1} q^{e-1}})^{q^{e-2}} \\ &= g^{x_0 q^{e-2}} \cdot g^{x_1 q^{e-1}} \cdot (g^{q^e})^{x_2 + x_3 q + \cdots + x_{e-1} q^{e-3}} \\ &= g^{x_0 q^{e-2}} \cdot g^{x_1 q^{e-1}} \end{aligned}$$

Recordemos que ya determinamos el valor de x_0 y el elemento $g^{q^{e-1}}$ tiene orden q en G. Para hallar x_1, tenemos que resolver el PLD

$$(g^{q^{e-1}})^{x_1} = (h \cdot g^{-x_0})^{q^{e-2}}$$

para x_1. De nuevo, aplicando el algoritmo conocido, podemos resolver esto en S_q pasos. Así en $\mathcal{O}(2S_q)$ pasos hemos determinado los valores x_0 y x_1 satisfaciendo

$$g^{(x_0 + x_1 q) q^{e-2}} = h^{q^{e-2}} \quad \text{en} \quad G.$$

Similarmente encontramos x_2 resolviendo el PLD

$$(g^{q^{e-1}})^{x_2} = (h \cdot g^{-x_0 - x_1 q})^{q^{e-3}},$$

y en general, una vez determinados $x_0, \cdots, x_{i-1}$ el valor de x_i se obtiene resolviendo

$$(g^{q^{e-1}})^{x_i} = (h \cdot g^{-x_0 - x_1 q - \cdots - x_{i-1} q^{i-1}})^{q^{e-i-1}} \quad \text{en} \quad G.$$

Cada una de estas es un PLD cuya base es de orden q así que cada una de ellas puede ser resuelta en S_q pasos. Así, luego de $\mathcal{O}(eS_q)$ pasos, obtenemos un exponente $x = x_0 + x_1 q + \cdots + x_{e-1} q^{e-1}$ satisfaciendo $g^x = h$, lo que resuelve el PLD original. ■

4. Factorización entera y RSA

4.1. La fórmula de Euler y raíces módulo pq

El método de intercambio de clave Diffie-Hellman y el CCP ElGamal estudiados anteriormente descansan en el hecho de que es fácil calcular potencias $a^n \ (mod\ p)$ pero difícil de recuperar el exponente n si sólo se conocen los valores de a y $a^n \ (mod\ p)$. Un resultado esencial que utilizamos para analizar la seguridad de Diffie-Hellman y ElGamal es el Pequeño Teorema de Fermat (PTF),

$$a^{p-1} \equiv 1 \ (mod\ p) \quad \text{para todo } a \not\equiv 0 \ (mod\ p).$$

Es natural preguntarse qué pasa si reemplazamos a p con un número m que no es primo. Algunos cálculos muestran que el teorema no sigue valiendo. En esta sección vamos a buscar una correcta generalización de este teorema cuando $m = pq$ es un producto de dos primos distintos, ya que este es el caso más importante para las implicaciones criptográficas.

El intercambio de clave Diffie-Hellman y el CCP ElGamal descansan para su seguridad en la dificultad de resolver las ecuaciones de la forma

$$a^x \equiv b \ (mod\ p),$$

donde a, b y p son cantidades conocidas, p es primo y x es desconocido. El CCP RSA, que estudiaremos en la siguiente sección, descansa en la dificultad de resolver las ecuaciones de la forma

$$x^e \equiv c \ (mod\ N),$$

donde ahora las cantidades e, c y N son conocidas, y x es desconocido. Es decir, la seguridad del RSA se basa en asumir la dificultad de tomar raíces e-simas módulo N. Nos preguntamos si es razonable asumir esto: Si N es primo, resulta relativamente fácil calcular dichas raíces, como describiremos en la siguiente proposición.

Proposición 4.1.1. *Sea p un primo y $e \geq 1$ un entero satisfaciendo que $MCD(e, p-1) = 1$. Sea d el inveso de e modulo $p-1$. Luego, la congruencia*

$$x^e \equiv c \ (mod\ p)$$

tiene única solución $x \equiv c^d \ (mod\ p)$.

Demostración. Si $c \equiv 0 \ (mod\ p)$, luego $x \equiv 0 \ (mod\ p)$ es la única solución y listo. Asumamos $c \not\equiv 0 \ (mod\ p)$. La prueba es una aplicación del PTF: la congruencia $de \equiv 1 \ (mod\ p-1)$ significa que existe un entero k tal que

$$de = 1 + k(p-1).$$

Ahora vamos a chequear que c^d es una solución de $x^d \equiv c \ (mod\ p)$:

$$\begin{array}{rcll} (c^d)^e & \equiv & c^{de} \ (mod\ p) & \text{por leyes de los exponentes,} \\ & \equiv & c^{1+k(p-1)} \ (mod\ p) & \text{ya que } de = 1 + k(p-1), \\ & \equiv & c \cdot (c^{p-1})^k \ (mod\ p) & \text{leyes de los exponentes,} \\ & \equiv & c \cdot 1^k \ (mod\ p) & \text{por el PTF,} \\ & \equiv & c \ (mod\ p). & \end{array}$$

Esto completa la prueba de que $x = c^d$ es una solución de $x^e \equiv c \ (mod\ p)$. Para ver la unicidad, supongamos que x_1 y x_2 son soluciones a la congruencia $x^e \equiv c \ (mod\ p)$. Hemos probado que $z^{de} \equiv z \ (mod\ p)$ para cualquier valor no nulo de z. Luego tenemos que

$$x_1 \equiv x_1^{de} \equiv (x_1^e)^d \equiv c^d \equiv (x_2^e)^d \equiv x_2^{de} \equiv x_2 \ (mod\ p).$$

Así, x_1 y x_2 son equivalentes módulo p. Luego, la ecuación tiene como única solución $x \equiv c^d \ (mod\ p)$. ■

La proposición anterior nos muestra que es fácil calcular las raíces e-simas si el módulo es un primo p, pero la situación para un módulo compuesto N es algo distinta. Si sabemos como factorizar N, es nuevamente sencillo calcular las raíces e-simas. La siguiente proposición explica cómo hacerlo si $N = pq$ con p y q primos distintos, y la prueba para el caso general la haremos en el apéndice.

Proposición 4.1.2. *Sean p y q primos distintos, y sea $e \geq 1$ satisfaciendo*

$$MCD(e, (p-1)(q-1)) = 1.$$

Sea d el inverso de e módulo $(p-1)(q-1)$, o sea

$$de \equiv 1 \ (mod\ (p-1)(q-1)).$$

Luego, la congruencia

$$x^e \equiv c \ (mod\ pq)$$

tiene una única solución $x \equiv c^d \ (mod\ pq)$.

Demostración. Haremos la demostración por casos:

1. Si $MCD(c, pq) = 1$: La prueba es casi idéntica a la anterior, pero en vez de usar el PTF, usamos la formula de Euler que vimos en los preliminares. La congruencia $de \equiv 1 \ (mod\ (p-1)(q-1))$ significa que existe un entero k tal que

   $$de = 1 + k(p-1)(q-1).$$

   Ahora veamos que c^d es solución de $x^e \equiv c \ (mod\ pq)$:

   $$\begin{array}{rcll} (c^d)^e & \equiv & c^{de} \ (mod\ pq) & \text{por leyes de los exponentes,} \\ & \equiv & c^{1+k(p-1)(q-1)} \ (mod\ pq) & \text{ya que } de = 1 + k(p-1)(q-1), \\ & \equiv & c \cdot (c^{(p-1)(q-1)})^k \ (mod\ pq) & \text{leyes de los exponentes,} \\ & \equiv & c \cdot 1^k \ (mod\ pq) & \text{por la fórmula de Euler,} \\ & \equiv & c \ (mod\ pq). & \end{array}$$

   Esto completa la prueba de que $x = c^d$ es solución. Veamos que es la única: Supongamos que $x = u$ es también una solución. Entonces

$$\begin{aligned} u &\equiv u^{de-k(p-1)(q-1)} \ (mod\ pq) && \text{ya que } de = 1 + k(p-1)(q-1), \\ &\equiv (u^e)^d \cdot (u^{(p-1)(q-1)})^{-k} \\ &\equiv (u^e)^d \cdot 1^{-k} \ (mod\ pq) && \text{usando la fórmula de Euler,} \\ &\equiv c^d \ (mod\ pq) && \text{pues } u \text{ era solución de la ecuación.} \end{aligned}$$

Luego toda solución de la ecuación es igual a c^d módulo pq, o sea que es la única solución.

2.

■

Observación 4.1.3. La proposición anterior nos da un algoritmo para resolver $x^e \equiv c\,(mod\,pq)$ que implica primero resolver $de \equiv 1\ (mod\ (p-1)(q-1))$ y luego computar $c^d\ (mod\ pq)$. Usualmente podemos computar más rápidamente usando un valor más pequeño de d. Sea $g = MCD(p-1, q-1)$ y supongamos que resolvemos la siguiente congruencia para d:

$$de \equiv 1 \ \left(mod\ \frac{(p-1)(q-1)}{g}\right).$$

La fórmula de Euler nos dice que $a^{(p-1)(q-1)/g} \equiv 1\ (mod\ pq)$. Así, como en la proposición anterior, si escribimos $de = 1 + k(p-1)(q-1)/g$, entonces

$$(c^d)^e = c^{de} = c^{1+k(p-1)(q-1)/g} = c \cdot (c^{(p-1)(q-1)/g})^k \equiv c\ (mod\ pq).$$

Luego, usando este valor más pequeño de d, $c^d\ (mod\ pq)$ sigue siendo solución de $x^e \equiv c\ (mod\ pq)$.

4.2. El CCP RSA

Ana y Bruno se enfrentan al usual problema de intercambiar información importante a través de medios inseguros. En la sección anterior vimos distintas formas de hacer esto, basadas en la dificultad del PLD. Ahora describiremos el Criptosistema de Clave Pública RSA (nombrado por sus creadores, Ron Rivest, Aid Shamir y Leonard Adleman), el primero en inventarse y el mejor CCP conocido. La seguridad del RSA reside en la siguiente dicotomía:

- **Preparación.** Sean p y q primos grandes, sea $N = pq$, y sean e y c enteros.
- **Problema.** Resolver la congruencia $x^e \equiv c\ (mod\ N)$ para la variable x.
- **Fácil.** Bruno, que conoce los valores de p y q, puede resolver fácilmentela ecuación con el algoritmo descripto anteriormente.
- **Difícil.** Inés, que no conoce los valores de p y q, no puede hallar x tan fácilmente.
- **Dicotomía.** Resolver $x^e \equiv c\ (mod\ N)$ es sencillo para una persona que posee información extra, pero aparentemente difícil para todos los demás.

El algoritmo es de la siguiente manera: La clave secreta de Bob son los primos grandes p y q. Su clave pública es el par (N, e) que consiste del producto $N = pq$ y de un exponente de encriptación e que sea coprimo con $(p-1)(q-1)$. Ana convierte su texto en un entero m entre 1 y N, y lo encripta computando la cantidad

$$c \equiv m^e \ (mod\ N)$$

. El entero c es el texto cifrado que le envía a Bruno. Luego, es una tarea sencilla para Bob resolver la ecuación $x^e \equiv c \ (mod\ N)$ para recuperar el mensaje m de Ana, pues conoce la factorización $N = pq$. En cambio para Inés, aunque pueda interceptar el mensaje c, a menos que sepa cómo factorizar N tiene presuntamente un trabajo difícil en resolver $x^e \equiv c \ (mod\ N)$.

Observación 4.2.1. Las cantidades N y e que forman la clave pública de Bruno se llaman, respectivamente, el *módulo* y el *exponente de encriptación*. El número d que Bruno usa para decriptar el mensaje de Ana, esto es el número d que satisface

$$ed \equiv 1 \ (mod\ N),$$

se llama *exponente de decriptación*. Está claro que la encriptación se puede hacer más eficientemente si el exponente de encriptación e es pequeño, y similarmente, que la decriptación es más eficiente si d es pequeño. Por supuesto, Bruno no los puede elegir ambos pequeños, ya que al fijar uno, el otro queda determinado por la ecuación que satisfacen. Notemos que Bruno no puede tomar $e = 2$, pues e debe ser coprimo con $(p-1)(q-1)$, así que el menor valor posible para e es 3. Hasta donde se sabe, tomar $e = 3$ es tan seguro como tomarlo más grande, pero todavía hay algunas dudas. Otra alternativa para Bruno es tomar d pequeño, y usar la congruencia para determinar e que resultará ser grande. Pero resulta que esto puede conducir a una versión insegura del RSA: para valores de d menores que $N^{1/4}$, la teoría de Fracciones Continuas permite a Inés romper el RSA.

Observación 4.2.2. La clave pública de Bob incluye el número $N = pq$, que es el producto de dos primos secretos p y q. De todos modos, si Inés conoce el valor de $(p-1)(q-1)$, puede resolver la ecuación $x^e \equiv c \ (mod\ N)$, y así decriptar el mensaje de Ana y Bruno. Expandiendo $(p-1)(q-1)$ obtenemos

$$(p-1)(q-1) = pq - p - q + 1 = N - (p+q) + 1$$

. Bruno ya publicó el valor de N, así que si Inés puede determinar el valor de la suma $p+q$, entonces tiene el valor de $(p-1)(q-1)$ que le permite decriptar los mensajes. De hecho, si Inés logra conocer los valores $p+q$ y pq, es fácil computar los valores de p y q, simplemente calculando las raíces del polinomio

$$X^2 - (p+q)X + pq$$

, ya que se factoriza como $(X-p)X-q)$. Así que una vez que Bruno publica el valor de $N = pq$, para Inés no es más difícil hallar el valor de $(p-1)(q-1)$ que hallar a p y q.

Observación 4.2.3. Hemos visto que no es más fácil para Inés determinar $(p-1)(q-1)$ de lo que es factorizar N. Pero esto no prueba que Inés deba factorizar N para decriptar los mensajes de Bruno. El punto es que lo que Inés necesita es resolver la ecuación $x^e \equiv c \ (mod\ N)$, y podría ser posible encontrar un algoritmo eficiente que la resuelva aún sin conocer el valor de $(p-1)(q-1)$, pero todavía nadie determinó si existe un tal método.

4.3. Cuestiones de implementación y seguridad

Si bien nuestro objetivo es analizar algunos aspectos de las matemática subyacente en los díficiles problemas de la criptografia moderna, no podemos dejar de hacer mención a algunas cuestiones relacionadas con la implementación y la seguridad. Debemos tener en claro, que no estamos ni cerca de la superficie de lo vasto que es este tema, pero daremos algunos ejemplos para mostrar que hay un largo trecho hasta desarrollar un sistema de comunicaciones seguro que va más allá de un criptosistema basado en problemas matemáicos.

Ejemplo 4.3.1. (Ataque de un intermediario) Supongamos que Inés más que solo espiar, tiene control sobre la red de comunicación entre Ana y Bruno. En este caso ella puede instituir, lo que se conoce como ataque deun intermediario. Daremos el ejemplo para el intercambio de clave Diffie Hellman:

Recordemos que en este caso, Ana le envía a Bruno el valor $A = g^a$ y Bruno le manda a Ana el valor $B = g^b$, donde los cálculos se realizan en $\mathbb{F}_p$. Inés elige su propio exponente e y calcula el valor $E = g^e$. Luego ella intercepta las comunicaciones entre Ana y Bruno y en vez de mandar A a Bruno y B a Ana, ella les manda a ambos el valor E. Supongamos que Ana y Bruno usan luego sus supuestos valores secretos que compartieron como clave para cifrar mensajes que se van a enviar. Por ejemplo, Ana encripta un mensaje m usando E^a como clave. Inés lo intercepta y lo puede desencriptar usando como clave A^e y leer el mensaje. Luego lo reencripta usando B^e como clave y se lo manda a Bruno. Como él puede desencriptarlo usando E^b él es inconciente de la brecha en la seguridad que se acaba de abrir.

Notemos que en este ataque, Inés no resolvió el problema subyacente, el PLD o el Diffie-hellman, pero puede leer las comunicaciones de Ana y Bruno y ellos no perciben el logro de Inés.

Ejemplo 4.3.2. Supongamos que Inés puede convencer a Ana para decriptar mensajes RSA al azar usando su clave privada (la de Ana). Esto es factible, pues una manera de Ana de autenticar su identidad como la propietaria de la clave pública (N, e) es mostrar que sabe como decriptar mensajes (decimos que Inés tiene acceso a un oráculo del RSA). Inés puede usar la generosidad de Ana de esta manera: Supongamos que Inés ha interceptado un texto cidrado que Bruno le mandó a Ana. Inés elige un valor k aleatorio y le envía a Ana el mensaje

$$c' \equiv k^e \cdot c \ (mod\ N).$$

Ana decripta c' y le devuelve el resultado m' a Inés, donde

$$m' \equiv (c')^d \equiv (k^e \cdot c)^d \equiv (k^e \cdot m^e)^d \equiv k \cdot m \ (mod\ N).$$

Como Inés conoce $k \cdot m \ (mod\ N)$ y conoce k, inmediatamente recupera el mensaje m original de Bruno. Así Inés obtuvo el mensaje de Bruno sin saber factorizar N, y así la dificultad del problema subyacente fue irrelevante.

Ejemplo 4.3.3. Supongamos que Ana publica dos diferentes exponentes e_1 y e_2 para usar con su modulo público, N, y así Bruno encripta un mensaje m usando ambos exponentes. Si Inés intercepta los mensajes

$$c_1 \equiv m^{e_1} \ (mod\ N) \quad \text{y} \quad c_2 \equiv m^{e_2} \ (mod\ N),$$

ella puede obtener una solución a la ecuación

$$e_1 \cdot u + e_2 \cdot v = \text{MCD}(e_1, e_2)$$

y usarla para calcular

$$c_1^u \cdot c_2^v \equiv (m^{e_1})^u \cdot (m^{e_2})^v \equiv m^{e_1 \cdot u + e_2 \cdot v} \equiv m^{MCD(e_1,e_2)} \ (mod\ N).$$

Si e_1 y e_2 fuesen coprimos, Inés recuperaría el mensaje de Bruno. Y más generalmente, puede hacerlo aún cuando Bruno use n exponentes si ellos son coprimos. La moraleja es que Ana debiera usar como mucho un solo exponente de encriptación para un módulo dado.

4.4. Testeo de Primalidad

Bruno ya entendió el algoritmo RSA y está listo para comunicarse con Ana usándolo. ¿O no lo está? Para crear una clave RSA, Bruno necesita dos primos *muy grandes* p y q. No es suficiente que elija dos números muy grandes pero posiblemente compuestos, pues si no son primos, deberá saber cómo factorizarlos para decriptar el mensaje. Pero peor aún, si p y q tienen factores primos pequeños, incluso Inés podría factorizar pq y romper el sistema de Bruno. Así que Bruno se enfrenta con la tarea de encontrar números primos grandes. Más precisamente, necesita una forma de distinguir entre números primos y números compuestos, ya que así podrá probar con números muy grandes hasta dar con un par de primos. Más adelante discutiremos sobre la probabilidad de que un número elegido al azar sea primo, pero por ahora es suficiente saber que tiene una posibilidad razonable de éxito. Por ejemplo, supongamos que Bruno elige el siguiente número grande:

$$n = 31987937737479355332620068643713101490952335301$$

y quiere saber si es primo. Primero prueba si lo dividen factores pequeños, pero encuentra que n no es divisible por ningún primo menor que 1000000. Así que empieza a sospechar que es primo, pero luego computa la cantidad 2^{n-1} módulo n y encuentra que

$$2^{n-1} \equiv 128126595355135906413360121624715183605316007 4 \ (mod\ n)$$

. Con lo cual, Bruno inmediatamente reconoce que n no es primo, aunque no tenga idea de cómo factorizarlo. Esto es gracias al PTF, que nos dice que si p es primo, $a^{p-1} \equiv 1 \ (mod\ p)$ (a menos que p divida a a). Antes de seguir, veamos una versión generalizada del PTF, que no tiene resricciones para el a.

Teorema 4.4.1. *Sea p un número primo. Entonces*

$$a^p \equiv a \ (mod\ p) \quad \textit{para todo entero } a$$

.

Demostración. Si $p \nmid a$, la primer versión del PTF dice que $a^{p-1} \equiv 1 \ (mod\ p)$. Multiplicando a ambos lados por a, obtenemos $a^p \equiv a \ (mod\ p)$. Por otro lado, si $p \mid a$, ambos lados de la equivalencia son 0 módulo p. ■

Volviendo con Bruno, supongamos que ahora elige el número

$$n = 296795298595169276282041874013832900431516513 1$$

. Después de chequear su divisibilidad por primos pequeños, computa 2^n módulo n y encuentra que

$$2^n \equiv 2 \ (mod\ n)$$

. ¡Pero esto no demuestra que n sea primo! El PTF funciona en una sola dirección,

$$\text{Si } p \text{ es primo, entonces } a^p \equiv a \ (mod\ p).$$

Pero nada dice que esa congruencia no pueda ser verdadera para un n compuesto. De hecho,

$$2^{341} \equiv 2 \ (mod\ 341) \ \text{ con } 341 = 11 \cdot 31$$

. De todos modos, en un vago sentido filosófico, el hecho de que $2^n \equiv 2 \ (mod\ n)$ hace más esperable que n sea primo, ya que si el valor de 2^n módulo n hubiese sido diferente, habríamos sabido que n era compuesto. Esto nos lleva a la siguiente definición.

Definición 4.4.2. Dado un entero n, decimos que un entero a es un *testigo para (la composición de)* n si

$$a^n \not\equiv a \ (mod\ n).$$

Como observamos antes, un solo testigo combinado con el PTF alcanza para probar sin niguna duda que n es compuesto. Así, una forma de testear la probabilidad de que un número n sea primo, sería probar con varios números, a_1, a_2, a_3, etc. Si alguno resulta un testigo para n, entonces Bruno sabrá que n es compuesto; y si ninguno de ellos es un testigo para n, entonces Bruno sospechará (aunque no lo sabrá a ciencia cierta) que n es primo. Desafortunadamente, esto no es tan cierto, ya que hay números, como el 561 que tienen una cierta particularidad: Son compuestos pero no tienen ningún testigo. Por ejemplo, $561 = 3 \cdot 11 \cdot 17$ pero

$$a^{561} \equiv a \ \text{ para todo entero } a.$$

Los enteros compuestos que no tienen testigos se llaman *números de Carmichael*, y se ha probado que aunque son raros, hay infinitos. Así que Bruno necesita algo más fuerte que el PTF para probar que un número es (probablemente) primo. La siguiente propiedad de los primos se usa para formular el *Test de Miller-Rabin*.

Proposición 4.4.3. *Sea p un primo impar y escribamos*

$$p - 1 = 2^k q \ \text{ con } q \text{ impar.}$$

Sea a cualquier número no divisible por p. Entonces una de las siguientes dos condiciones es verdadera:

(I) a^q es congruente con 1 módulo p.

(II) Alguno de los siguientes a^q, a^{2q}, a^{4q}, $\cdots$, $a^{2^{k-1}q}$ es congruente con -1 módulo p.

Demostración. El PTF nos dice que $a^{p-1} \equiv 1 \ (mod\ p)$. Esto significa que si miramos la lista de números

$$a^q,\ a^{2q},\ a^{4q},\ \cdots,\ a^{2^{k-1}q},\ a^{2^k q},$$

sabemos que el último número de la lista, que es igual a a^{p-1}, es congruente con 1 módulo p. Además, cada número en la lista es el cuadrado del anterior. Así que una de las siguientes dos posibilidades debe ocurrir:

(I) El primer número de la lista es congruente a 1 módulo p.

(II) Algún número de la lista no es congruente a 1 módulo p, pero cuando se lo eleva al cuadrado resulta congruente a 1 módulo p. Pero el único número que satisface simultáneamente

$$b \not\equiv 1 \ (mod\ p) \ \text{ y } \ b^2 \equiv 1 \ (mod\ p)$$

es -1, así que uno de los números de la lista es congruente con -1 módulo p.

Esto completa la demostración. ■

Definición 4.4.4. Sea n un número impar, y escribamos $n-1 = 2^k q$ con q impar. Un entero a satisfaciendo $MCD(a, n) = 1$ se llama *testigo de Miller-Rabin para (la composición de) n* si las siguientes dos condiciones son verdaderas:

(I) $a^q \not\equiv 1 \ (mod\ n)$.

(II) $a^{2^i q} \not\equiv -1 \ (mod\ n)$ para ningún $i = 0,\ 1,\ 2, \cdots, k-1$.

Por la proposición anterior, si existe un a que sea testigo de Miller-Rabin para n, entonces n es definitivamente compuesto. Esto nos lleva al algoritmo de Miller-Rabin para identificar números compuestos.

Ahora, supongamos que Bruno quiere chequear si un número n es probablemente primo. Para hacerlo, usa el algoritmo de Miller-Rabin con una serie de valores de a seleccionados al azar. ¿Por qué es mejor que el PTF? La respuesta es que no hay números como los de Carmichael para el test de Miller-Rabin, y de hecho, todo número compuesto tiene muchos testigos de Miller-Rabin:

Proposición 4.4.5. *Sea n un número compuesto. Entonces al menos el 75 % de los números entre 1 y $n-1$ son testigos de Miller-Rabin para n.*

Ahora, Bruno toma un número grande n y le realiza el test de Miller-Rabin para, digamos, 10 valores de a. Si alguno de esos valores es un testigo de Miller-Rabin para n, Bruno sabe que n es compuesto. Pero supongamos que ninguno lo es; la proposición anterior dice que, si n es compuesto, cada vez que Bruno prueba con un valor de a, tiene 75 % de chances de que sea un testigo. Como en 10 intentos Bruno no encontró ningún testigo, es razonable concluir que la probabilidad de que n sea compuesto es como mucho $(25\,\%)^{10}$ que es aproximadamente 10^{-6} (luego se verá con más precisión este argumento en términos probabilísticos). Lógicamente, cuantos más valores de a pruebe que no sean testigos para n, aumenta la probabilidad de que n sea primo.

4.4.1. Distribución de los números primos

Si Bruno eligiera un número natural al azar ¿Qué chances tiene de que sea primo? Para responder esta pregunta vamos a necesitar un teorema de la teoría de números muy importante, que usa esta definición:

Definición 4.4.6. Para cualquier número x, sea

$$\pi(x) = (\text{ cantidad de primos p tales que } 2 \leq p \leq x).$$

Teorema 4.4.7. *(El teorema del número primo)*

$$\lim_{x\to\infty} \frac{\pi(x)}{x/lm(x)} = 1$$

Para propositos criptograficos necesitaremos primos grandes, por ej. de 300 cifras, o casi equivalentemente, primos con una longitud de 1024 bits, pues $2^{1024} \simeq 10^{308,25}$. Así para estimar cúantos primos hay entre 2^{1023} y 2^{1024}, usamos el teorema anterior y nos da la siguiente aproximación:

$$\pi(2^{1024}) - \pi(2^{1023}) \simeq 2^{1013,53}$$

Luego hay muchos primos en este intervalo. Intuitivamente, este teorema dice que entre 1 y x, la proporción de números que son primos es aproximadamente $1/ln(x)$. Dicho de otra manera, el teorema afirma que

Un número N elegido tiene probabilidad $1/ln(N)$ de ser primo.

Por supuesto, que esta afirmación no tiene mucho sentido, un número elegido o es primo o no lo es; no puede ser parcialmente primo. Una mejor interpretación de este enunciado, es que describe cúantos primos uno espera encontrar en un intervalo alrededor de N: Fijados dos números positivos c_1 y c_2, tales que $c_1 < c_2$. Bruno elige al azar un número n en el intervalo $c_1N \leq n \leq c_2N$. sea

$$P(c_1, c_2, N) = [\text{Probabilidad de que un entero en el intervalo } c_1N \leq n \leq c_2N \text{ sea primo}].$$

Luego haciendo un cambio de variables y usando el teorema de los números primos tenemos

$$\lim_{N\to\infty} \frac{P(c_1, c_2, N)}{(c_2 - c_1)/ln(N)} = \lim_{N\to\infty} \frac{(\pi(c_2N) - \pi(c_1N))\ ln(N)}{(c_2 - c_1)N} = 1$$

Observación 4.4.8. Existen muchas cuestiones abiertas en cuanto a la distribución de los números primos, de las cuales la más famosa e importante es la hipótesis de Riemann. El enunciado usual de la hipótesis de Riemann requiere algo de análisis complejo: La función zeta de Riemann $\varsigma(s)$ esta definida por la serie

$$\varsigma(s) = \sum_{n=1}^{\infty} \frac{1}{n^s},$$

que converge cuando s tiene parte real mayor a 1. Esta función tiene una prolongación analítica a la función entera con un polo simple en $s = 1$ y ningún otro más. La hipótesis de Riemann dice que si $\varsigma(\sigma + it) = 0$ con σ y t reales, $0 \leq \sigma \leq 1$, entonces $\sigma = \frac{1}{2}$. A priori, puede parecer que este enunciado no tiene relación con los números primos, sin embargo se puede probar que $\varsigma(s)$ es equivalente al producto

$$\varsigma(s) = \prod_{p \ primos} \left(1 - \frac{1}{p^s}\right)^{-1}$$

que brinda información sobre el conjunto de los números primos. Hay muchos enunciados sobre los números primos equivalentes a la hipótesis de Riemann, por ejemplo, (si recordamos el teorema de los números primos) se puede probar que la hipótesis de Riemann es equivalente a que

$$\pi(x) = \int_2^x \frac{dt}{ln\ t} + \mathcal{O}\left(\sqrt{x} \cdot ln(x)\right).$$

Esta fórmula es más fuerte que el teorema de los números primos ya que esta integral es aproximadamente $x/ln(x)$.

4.4.2. Pruebas de primalidad versus test probabilísticos

El test de Miller-Rabin es un método muy poderoso y práctico para encontrar números muy grandes que son posibles primos. Sabemos que cualquier número compuesto tiene muchos testigos de Miller-Rabien, luego 50 o 100 repeticiones de este test dan evidencia sólida de que un n es primo. Sin embargo, esto no es una prueba rigurosa. Supongamos que Bruno quiere estar completamente seguro de que el número n que elige es primo. En principio, Bruno puede chequear si los números 1,2,3,4,... hasta $\sqrt{n}$ dividen a n. Si ninguno lo hace, con certeza n es primo. Desafortunadamente, si n es grande, digamos del orden de 2^{1000}, el Sol se apagaría antes de que esta tarea esté terminada. Notemos que este método es del orden de $\sqrt{n}$, es decir, es de tiempo exponencial según nuestra definición. Sería conveniente si pudiesemos usar test de Miller-Rabin para eficiente y conclusivamente probar que un número es primo. Más precisamente, buscaríamos un algoritmo de tiempo polinómico. que prube la primalidad. Si se aceptase una versión generalizada de la hipótesis de Riemman, la siguiente proposición dice que esto puede hacerse.

Proposición 4.4.9. *Si una versión generalizada de la hipótesis de Riemman es verdadera, luego cada número compuesto n tiene un testigo de Miller-Rabin a.*

4.5. El algoritmo de factorización p-1 de Pollard

Vimos anteriormente que es relativamente fácil chequear si un número grande es (probablemente) primo. Esto es bueno, ya que el criptosistema RSA requiere de primos grandes para operar. Por el contrario, la seguridad del RSA reside en la aparente dificultad de factorizar números grandes. El estudio de la factorización se remonta a la antigua Grecia (por lo menos), pero fue recién con el advenimiento de las computadoras que la gente empezó a desarrollar algoritmos capces de factorizar números muy grandes. La paradoja del RSA es que para hacerlo más eficiente, queremos usar un módulo $N = pq$ que sea lo más pequeño posible; por otro lado, si un oponente puede factorizar N, nuestros mensajes encriptados ya no serán seguros. Es entonces vital entender qué tan difícil es factorizar números grandes, y en particular, entender las capacidades de los diferentes algoritmos de factorización actualmente utilizados. Comenzaremos estudiando un algoritmo llamado *Método $p-1$ de Pollard*. Si bien no es útil para todos los números, hay ciertos números para los cuales es bastante eiciente. El método de Pollard demuestra que hay módulos RSA que son inseguros aunque a primera vista parezcan seguros. Nos dan un número $N = pq$ y nuestra tarea es determinar los factores primos p y q. Supongamos que por algún motivo logramos encontrar un entero L con la propiedad de que

$$p-1 \text{ divide a } L \quad \text{y} \quad q-1 \text{ no divide a } L$$

Esto signiica que hay enteros i, j y k con $k \neq 0$, satisfaciendo

$$L = i(p-1) \quad \text{y} \quad L = j(q-1) + k.$$

Consideremos qué pasa si tomamos al azar un entero a y computamos a^L. Asumiremos que $p \nmid a$ y $q \nmid a$, ya que al ser p y q grandes, este será probablemente el caso. Además, si $p \mid a$ y $q \nmid a$, podemos recuperar $p = MCD(a, N)$. Por el PTF,

$$\begin{array}{rcccccccl} a^L & = & a^{i(p-1)} & = & (a^{p-1})^i & \equiv & 1^i & \equiv & 1 \pmod{p}, \\ a^L & = & a^{j(q-1)+k} & = & a^k \cdot (a^{q-1})^j & \equiv & a^k \cdot 1^j & \equiv & a^k \pmod{q}. \end{array}$$

El exponente k es distinto de cero, así que es bastante improbable que a^k sea congruente a 1 módulo q. Así, con muy alt probabilidad, es decir para la mayoría de las elecciones de a, obtendremos que

$$p \text{ divide a } a^L - 1 \quad \text{y} \quad q \text{ no divide a } a^L - 1.$$

Pero esto significa que podemos recuperar p computando $p = MCD(a^L - 1, N)$. Esto está muy bien, pero ¿cómo podemos encontrar ese tal L? La observación de Pollard es que si $p-1$ resulta ser un producto de primos pequeños, entonces va a dividir a $n!$ ara algunos valores no muy grandes de n. Entonces la idea es: para cada $n = 2, 3, 4, ...$ elegimos un a y computamos

$$MCD(a^{n!} - 1, N).$$

Si el MCD es igual a 1, pasamos al siguiente valor de n. Si el MCD en algún momento es igual a N, tuvimos mala suerte, pero otro valor de a seguramente funcione. Y si obtenemos un número estrictamente entre 1 y N, ya encontramos un factor no trivial de N.

Observación 4.5.1. Hay dos observaciones importantes para hacer antes de poner en práctica la idea de Pollard. Primero en relación a la cantidad $a^{n!}-1$. Incluso para $a = 2$ y moderados valores de n, por ejemplo $n = 100$, es imposible computar $a^{n!} - 1$ exactamente, de hecho el número 2^{100} tiene más de 10^{157} dígitos, que es más que el número de partículas elementales del universo. Por suerte, no tenemos que computarlo exactaente, sólo nos interesa el máximo común divisor entre $a^{n!} - 1$ y N, así que alcanza con calcular $a^{n!} - 1$ $(mod\ N)$ y luego tomar el MCD con N. Segundo, ni siquiera necesitamos computar el valor $n!$, en cambio, asumiendo que calculamos $a^{n!}$ $(mod\ N)$ en el paso anterior, podemos computar el siguiente como

$$a^{(n+1)!} \equiv (a^{n!})^{n+1} \pmod{N}.$$

Observación 4.5.2. Notemos que es fácil para Ana y Bruno evitar los peligros del método $p-1$ de Pollard cuando crean sus claves RSA: basta con chequear que sus primos secretos p y q tienen la propiedad que ni $p-1$ ni $q-1$ se factoriza como producto de primos pequeños. Desde un punto de vista criptográfico, la importancia del método de Pollard reside en la siguiente lección: mucha gente podría no esperarse que la factorización de $p-1$ o $q-1$ tenga algo que ver con la factorización de pq. La moraleja es que aún si construimos uncriptosistema basado en un problema aparentemente difícil como la factorización entera, debemos estar atentos a los casos especiales que por razones poco obvias son más fáciles de resolver que el caso general.

Observación 4.5.3. Todavía no discutimos la probabilidad de que el algoritmo tenga éxito. Supongamos que p y q son primos, elegidos al azar, de más o menos el mismo tamaño. El método de Pollard funciona si al menos uno de $p-1$ o $q-1$ se factoriza como producto de primos pequeños. Claramente $p-1$ es par, así que podemos sacar un factor 2, pero luego, la cantidad $\frac{1}{2}(p-1)$ debería comportarse más o menos como un número aleatorio de tamaño aproximadamente $\frac{1}{2}p$. Lo que nos lleva a la siguiente pregunta:

¿Cuál es la probabilidad de que un entero elegido al azar, de tamaño aproximadamente n, divida a B!?

Notemos que, en particular, si n divide a $B!$, entonces cada primo l que divida a n debe satisfacer que $l \leq B$. Un número cuyos factores primos son todos menores o iguales que B se llaman *números B-suaves*. Es natural preguntarnos por la probabilidad de que un número elegido al azar de tamaño aproximadamente n sea un número B-suave. Dando vuelta la pregunta:

Dado n, ¿qué tan grande tiene que ser B para que un número elegido al azar de tamaño aproximadamente n tenga alta probabilidad de ser un número B-suave?

Estudiaremos esto en secciones posteriores.

4.6. Factorización vía diferencia de cuadrados

Los métodos de factorización más poderosos conocidos descansan en una de las identidades más simples en todas las matemáticas

$$X^2 - Y^2 = (X + Y)(X - Y)$$

Su potencial aplicabilidad en la factorización es inmediata. Para factorizar un número N, buscamos un entero b tal que la cantidad $N+b^2$ sea un cuadrado perfecto, digamos a^2. Luego tenemos

$$N = a^2 - b^2 = (a+b)(a-b),$$

y así efectivamente factorizamos N.

Si N es muy grande, no es muy plausible que un valor elegido al azar de b haga de $N+b^2$ un cuadrado perfecto. Necesitamos una manera más clara de elegir ese b. Una observación importante es que no necesariamente hay que escribir N es sí mismo como diferencia de cuadrados. A veces es suficiente escrimir algún múltiplo kN de N como diferencia de cuadrados, pues si

$$kN = a^2 - b^2 = (a+b)(a-b),$$

luego hay una chance razonable de que los factores de N estén sepaardos por los facores del lado derecho de la ecuación, i.e., que N tiene un factor no trivial en común con $a+b$ y $a-b$. Luego hay una manera simple de recuperar los factores calculando MCD$(N, a+b)$ y MCD $(N, a-b)$.

Los múltiplos de N son congruentes a 0 módulo N, luego en vez de buscar una diferencia de cuadrados $a^2 - b^2$ que sea múltiplo de N, buscamos números distintos a y b tales que

$$a^2 \equiv b^2 \ (mod\ N).$$

Esto es exactamente el mismo problema, por supuesto, pero el uso de la aritmética modular ayuda a clarificar nuestra tarea. En la prática no es feasible buscar directamente estos enteros a y b. En vez de eso, usamos un proceso de tres pasos que describiremos en una tabla. Este procedimiento, en una forma u otra, es la base de los métodos más modernos de factorización:

Construcción de la propiedad: Encontrar enteros $a_1, a_2, a_3, \cdots, a_r$ cumpliendo que la cantidad $c_i \equiv a_i^2 \ (mod\ N)$ se factorice como producto de primos chicos.
Eliminación: Tomar el producto $c_{i_1} c_{i_2} \cdots c_{i_s}$ de algunos de los c_i de forma tal que todo primo que aparezca en el producto aparezca a una potencia par. Luego, $c_{i_1} c_{i_2} \cdots c_{i_s} = b^2$ es un cuadrado perfecto.
Cálculo del MCD: Sea $a = a_{i_1} a_{i_2} \cdots a_{i_s}$ se calcula el máximo común divisor $d = MCD(N, a-b)$. Como

$$a^2 = (a_{i_1} a_{i_2} \cdots a_{i_s})^2 \equiv a_{i_1}^2 a_{i_2}^2 \cdots a_{i_s}^2 \equiv c_{i_1} c_{i_2} \cdots c_{i_s} \equiv b^2 \ (mod\ N),$$

hay una chance razonable de que d sea un factor no trivial de N.

Del paso 3, en realidad no hay mucho que decir, pues el algoritmo de Euclides nos dice que el MCD$(N, a-b)$ se puede calcular eficientemente en $\mathcal{O}(ln\ N)$ pasos. Por otra parte hay tanto que decir acerca del paso 1 que pospondremos su discusión hasta la próxima sección. Ahora analizaremos el paso 2:

Supondremos que cada uno de los números $a_1, \cdots, a_r$ hallados en el paso 1 tiene la propiedad de que $c_i \equiv a_i^2 \ (mod\ m)$ se factoriza en primos pequeños (digamos que cada c_i es un producto de primos elegidos del conjunto de los primeros t primos $\{p_1, p_2, \cdots, p_t\}$. Significa que existen exponentes e_{ij} tales que

$$\begin{aligned} c_1 &= p_1^{e_{11}} p_2^{e_{12}} p_3^{e_{13}} \cdots p_t^{e_{1t}}, \\ c_2 &= p_1^{e_{21}} p_2^{e_{22}} p_3^{e_{23}} \cdots p_t^{e_{2t}}, \\ &\vdots \qquad\qquad \vdots \\ c_r &= p_1^{e_{r1}} p_2^{e_{r2}} p_3^{e_{r3}} \cdots p_t^{e_{rt}}, \end{aligned}$$

Nuestra meta es tomar un producto de ciertos c_i de manera que cada primo del lado derecho de la ecuación aparezca a una potencia par. En otras palabras, nuestro problema se reduce a encontrar $u_1, u_2, \cdots, u_r \in \{0, 1\}$ tales que

$$c_1^{u_1} \cdot c_2^{u_2} \cdots c_r^{u_r} \text{ sea un cuadrado perfecto.}$$

Escribiendo el producto en terminos de la factorización prima tenemos

$$\prod_{i=1}^{r} c_i^{u_i} = \prod_{j=1}^{t} p_j^{\sum_{i=1}^{r} e_{ij} u_i}$$

En cualquier caso, tenemos que elegir los u_i de forma que los exponentes de los p_j sean pares.

Recapitulando, tenemos enteros

$$e_{11}, e_{12}, \cdots, e_{1t}, e_{21}, e_{22}, \cdots, e_{2t}, \cdots, e_{r1}, \cdots, e_{rt}$$

y buscamos enteros $u_1, u_2, \cdots, u_r$ tales que

$$\begin{aligned} e_{11}u_1 + e_{21}u_2 + \cdots + e_{r1}u_r &\equiv 0 \pmod 2, \\ e_{12}u_1 + e_{22}u_2 + \cdots + e_{r2}u_r &\equiv 0 \pmod 2, \\ \vdots \qquad\qquad &\vdots \\ e_{1t}u_1 + e_{2t}u_2 + \cdots + e_{rt}u_r &\equiv 0 \pmod 2, \end{aligned}$$

Este sistema de congruencias es un simple sistema de ecuaciones lineales sobre el cuerpo finito $\mathbb{F}_2$. Así técnicas básicas de algebra lineal como la eliminación gaussiana pueden ser usados para resolverlo. Sin embargo, en caso de querer factorizar un N muy grande, el conjunto de primos p_j deberá contener hasta incluso millones de primos, y el sistema tendrá millones de ecuaciones lineales que, incluso en el cuerpo $\mathbb{F}_2$, puede ser muy díficil de resolver sistemas de este tamaño. Pero, se ve que la mayoría de los coeficientes de los sistemas usados en la factorización son bastante escasos, es decir, que muchos de sus coeficientes son ceros. Por suerte, hay técnicas especiales para resolver sistemas lineales escasos más eficientes que la eliminación gaussiana

4.7. Números suaves, tamices, y relaciones de construcción para factorizaciones

En esta sección vamos a describir los dos métodos más rápidos conocidos para resolver los problemas de factorización más complicados, es decir, aquellos en los que hay que factorizar números de la forma $N = pq$ donde p y q son primos de aproximadamente el mismo orden

de magnitud. Empezaremos hablando de los números suaves, que forman la herramienta esencial para las relaciones de construcción. Luego describiremos el tamiz cuadrático, que es un método rápido para encontrar los números suaves necesarios. Finalmente, describiremos brevemente el tamiz del cuerpo de números, similar al cuadrático en que provee un método rápido para encontrar números suaves de una cierta forma. Sin embargo, cuando N es extremadamente grande, el tamiz del cuerpo de números es más rápido que el cuadrático, porque trabajando en un anillo más grande que $\mathbb{Z}$, usa números auxiliares más pequeños en su búsqueda de números suaves.

4.7.1. Números suaves

El paso de construcción de relación en el procedimiento de factorización descripto en la sección anterior, requiere que encontremos varios enteros con la propiedad de que $a^2 \bmod N$ se factorice como el producto de primos pequeños.

Definición 4.7.1. Un entero n se die B - suave, si todos los primos en su factorización son menores o iguales a B.

Definición 4.7.2. La función $\psi(X, B)$ se define como

$$\psi(X, B) = \text{Cantidad de números } B \text{ - suaves menores o iguales a } X.$$

Para poder evaluar la eficiencia del método de factorización en tres pasos necesitamos entender cómo se comporta $\psi(X, B)$ para valores grandes de X y B. Resulta que para obtener resultados útiles, las cantidades B y X deben crecer juntas de la manera correcta. Un teorema importante en este sentido fue probado por Canfield, Erdös y Pomerance.

Teorema 4.7.3. *(Canfield, Erdös, Pomerance) Fijar un número $0 < \varepsilon < 1$, y sean X y B que crecen al mismo tiempo satisfaciendo*

$$(lnX)^{\varepsilon} < lnB < (lnX)^{1-\varepsilon}.$$

Por conveniencia notacional, escribiremos $u = \dfrac{lnX}{lnB}$. Entonces

$$\psi(X, B) = X \cdot u^{-u(1+o(1))}.$$

Observación 4.7.4. Hemos usado la notación $o(1)$ por primera vez. Esta expresión denota una función que tiende a 0 cuando X tiende a infinito. Más generalmente, escribiremos

$$f(X) = o(g(X))$$

si el cociente $f(X)/g(X)$ tiende a 0 cuando X tiende a infinito.

La pregunta sigue siendo cómo deberíamos elegir B en términos de X. Y resulta que lo que necesitaremos es la siguiente función $L(X)$:

$$L(X) = e^{\sqrt{(lnX)(lnlnX)}}.$$

Luego, como una consecuencia inmediata del teorema anterior, obtenemos una estimación fundamental para ψ.

Corolario 4.7.5. *Para cualquier valor fijo de c con $0 < c < 1$,*

$$\psi(X, L(X)^c) = X \cdot L(X)^{-(1/2c)(1+o(1))} \quad \textit{cuando} \quad X \longrightarrow \infty.$$

Demostración. Notemos que si $B = L(X)^c$ y si tomamos cualquier $\varepsilon < 1/2$, entonces

$$lnB = clnL(X) = c\sqrt{(lnX)(ln\ lnX)}$$

satisface que $(lnX)^{\varepsilon} < lnB < (lnX)^{1-\varepsilon}$. Entonces podemos aplicar el teorema anterior con

$$u = \frac{lnX}{lnB} = \frac{1}{c} \cdot \sqrt{\frac{lnX}{ln\ lnX}}$$

para inferir que $\psi(X, L(X)^c) = X \cdot u^{-u(1+o(1))}$. Es fácil ver que este valor de u satisface

$$u^{-u(1+o(1))} = L(X)^{-(1/2c)(1+o(1))},$$

lo que completa la prueba. ■

Supongamos que deseamos factorizar N buscando valores a^2 $(mod\ N)$ que sean B-suaves. Para poder realizar el paso de eliminación de ecuaciones lineales necesitamos (al menos) tantos números B-suaves como primos menores que B. Esto es asi porque en el paso de eliminación los números suaves corresponden a las variables mientras que los primos menores que B corresponden a las ecuaciones, y necesitamos más variables que ecuaciones. Para asegurnos que este sea el caso, necesitamos que haya al menos $\pi(B)$ números B-suaves. Resultará que podemos tomar $B = L(N)^c$ para un valor adecuado de c. En la siguinte proposición usaremos el teorema de los números primos y la fórmula para $\psi(X, L(X)^c)$ dada por el corolario anterior para elegir el valor de c más pequeño que nos de alguna posiblidad de factorizar N usando este método.

Proposición 4.7.6. *Sea $L(X) = e^{\sqrt{(LnX)(LnLnX)}}$ como en el corolario anterior, sea N un entero grande y fijemos $B = L(N)^{1/\sqrt{2}}$.*

(a) Esperamos chequear aproximadamente $L(N)^{\sqrt{2}}$ números aleatorios módulo N para encontrar $\pi(B)$ números que sean B-suaves.

(b) Esperamos chequear aproximadamente $L(N)^{\sqrt{2}}$ números aleatorios de la forma a^2 módulo N para encontrar suficientes números B-suaves para factorizar N.

Demostración. Por el comentario anterior sabemos que (a) y (b) son equivalentes. Probemos (a): La probabilidad de que un número módulo N elegido al azar sea B-suave es $\psi(N, B)/N$. Para encontrar $\pi(B)$ números que sean B suaves, necesitamos chequear aproximadamente

$$\frac{\pi(B)}{\psi(N, B)/N} \quad \text{números.}$$

Queremos elegir un B que minimice esta función, ya que chequear la suavidad de un número es un proceso que consume mucho tiempo. El corolario anteior dice que

$$\psi(N, L(N)^c)/N \approx L(N)^{-1/2c},$$

así que fijamos $B = L(N^c)$ y buscamos el valor de c que minimice la función. El teorema de los números primos nos dice que $\pi(B) \approx B/Ln(B)$, asi que

$$\frac{\pi(L(N)^c)}{\psi(N, L(N)^c/N} \approx \frac{L(N)^c}{cLnL(N)} \cdot \frac{1}{L(N)^{-1/2c}} = L(N)^{c+1/2c} \cdot \frac{1}{cLnL(N)}.$$

El factor $L(N)^{c+1/2c}$ domina la última expresión así que elegimos el valor de c que minimice la cantidad $c+\frac{1}{2c}$. Esto es un cálculo elemental, se minimiza cuando $c = \frac{1}{\sqrt{2}}$ y el valor mínimo es $\sqrt{2}$. Así si elegimos $B \approx L(N)^{1/\sqrt{2}}$, entonces necesitamos chequear aproximadamente $L(N)^{\sqrt{2}}$ valores para encontrar $\pi(B)$ números que sean B-suaves, y por lo tanto para encontrar suficientes relaciones para factorizar N. ■

Observación 4.7.7. Cuando estimamos el esfuerzo necesario para factorizar N ignoramos por completo el trabajo requerido para chequear si un número dado es B-suave. Por ejemplo si chequeamos la B-suavidad usando intentos de división (i.e., dividir por cada primo menor que B) nos tomaría aproximadamente $\pi(B)$ divisiones. Teniendo esto en cuenta, la aproximación que hicimos en la proposición anterior se vuelve bastante ineficiente. Es por esto que describiremos un método más eficiente para generar números B-suaves, llamado el tamiz cuadrático.

4.7.2. El tamiz cuadrático

En esta sección agragermos la pieza final del rompecabezas que debemos resolver para factorizar grandes números vía el método de la diferencia de cuadrados descripto anteriormente:

¿Cómo podemos encontrar eficientemente muchos números $a < \sqrt{N}$ tales que cada a^2 $(mod\ N)$ sea B-suave?

Ya vimos que necesitamos tomar $B \approx L(N)^{1/\sqrt{2}}$ para tener buenas chances de factorizar N. Un primer acercamiento para encontrar cuadrados B-suaves módulo N era buscar fracciones $\frac{a}{b}$ tan cercanas como sea posible a $\sqrt{kN}$ para $k = 1, 2, ...$ Luego

$$a^2 \approx b^2 kN$$

así que a^2 $(mod\ N)$ es razonablemente pequeño, y así, es más esperable que sea B-suave. Un acercamiento alternativo que resulta ser mucho más rápido en la práctica es permitir valores un poco más grandes de a, y luego usar un proceso eficiente de cancelación llamado *tamiz* para simultáneamente crear muchos valores a^2 $(mod\ N)$ que sean B-suaves. Lo que veremos ahora es el *tamiz cuadrático* de Pomerance, que todavía es el método más rápido conocido para factorizar números grandes $N = pq$. Comenzaremos con un problema más simple: encontrar rápidamente bastantes números B-suaves menores que alguna cota X, sin preocuparnos de que tengan la forma a^2 $(mod\ N)$. Para hacer esto, adaptamos la *riba de Eratosthenes*, que es un método de la antigua Grecia para hacer listas de números primos. La idea de Eratosthenes para encontrar primos era la siguiente: empezamos marcando el primer primo, 2, y luego tachando todos los múltiplos de 2 mayores; luego marcamos el

siguiente número, 3 (que debe ser primo), y tachamos todos los múltiplos de 3 mayores; el menor número sin marcar es el 5, así que marcamos el 5, y tachamos todos los múltiplos de 5 mayores; y así sucesivamente. Al finalizar, los números marcados son primos. Notemos que algunos números serán tachados varias veces: por ejemplo, el 6, el 12 y el 18 se tacharán dos veces; el 30 y el 42 se tacharán tres, y así. Supongamos que ahora, en vez de tachar, dividimos. Esto es, comenzamos marcando el 2 y dividiendo por 2 a todos los múltiplos de 2; luego dividimos por 3 a todos los múltiplos de 3; por 5 a todos los múltiplos de 5; y así siguiendo. Si hacemos esto para todos los primos menores que B, ¿qué números terminan siendo divididos hasta llegar a 1? La respuesta es que son aquellos que son producto de primos distintos menores que B, o sea que en particular, ¡son B-suaves! Con lo cual terminamos con una lista de números B-suaves. Lamentablemente nos perdemos algunos B-suaves, digamos aquellos que son divisibles por productos de primos pequeños. Pero es fácil solucionar esto, luego de dividir por 3, y antes de dividir por 5, dividimos por 4: para hacer esto, sólo cancelamos otro factor 2 a cada múltiplo de 4 (pues ya lo habíamos dividido por 2 en el paso correspondiente). De esta forma, al finalizar, los número B-suaves menores que X son precisamente los números que fueron reducidos a 1. Sin embargo, nuestro objetivo no es hacer una lista de 1 a X que sean B-sueaves, lo que necesitamos es una lista de números de la forma a^2 $(mod\ N)$ que sean B-suaves. Nuestra estrategia para lograrlo, usa el polinomio

$$F(T) = T^2 - N.$$

Queremos empezar con un valor de a que sea un poco mayor a $\sqrt{N}$, así que fijamos

$$a = \lfloor \sqrt{N} \rfloor + 1,$$

donde $\lfloor x \rfloor$ denota el mayor entero menor o igual que x. Miramos a la lista de números

$$F(a),\ F(a+1),\ F(a+2), \cdots, F(b).$$

La idea es encontrar los números B-suaves en esta lista, "tamizando"los primos menores que B y viendo que números en la lista son "tamizados"hasta 1. Elegimos B lo suficientemente grande para que, al final del proceso de tamizado, sea esperable que hayamos encontrado suficientes números B-sueaves para factorizar N. La siguiente definición es útil para describir este proceso:

Definición 4.7.8. El conjunto de primos menores que B (o a veces el conjunto de potencias de primos menores que B) se llama la *base de factorización*.

Supongamos que p es un primo de nuestra base de factorización. ¿Qué números de la lista $F(a),\ F(a+1),\ F(a+2), \cdots, F(b)$, son divisibles por p? Equivalentemente, ¿qué números t entre a y b satisfacen

$$t^2 \equiv N\ (mod\ p)?$$

Si esta congruencia no tiene solución, descartamos el primo p, ya que p no divide ningún número de la lista. Caso contrario la congruencia tiene dos soluciones, que denotamos por

$$t = \alpha_p \quad \text{y} \quad t = \beta_p.$$

(Si $p = 2$, hay una sola solución α_p). Se sigue que cada uno de los números

$$F(\alpha_p),\ F(\alpha_p + p),\ F(\alpha_p + 2p),\ F(\alpha_p + 3p), \cdots$$

y cada uno de los números

$$F(\beta_p),\ F(\beta_p + p),\ F(\beta_p + 2p),\ F(\beta_p + 3p), \cdots$$

son divisibles por p. Así, podemos tamizar un factor p de cada p-ésima entrada en la lista original, comenzando con el menor valor de a satisfaciendo $a \equiv \alpha_p\ (mod\ p)$, y similarmente podemos tamizar un factor p de cada p-ésima entrada en la lista original, comenzando con el menor valor de a satisfaciendo $a = \beta_p\ (mod\ p)$.

Observación 4.7.9. Si p es un primo impar, la congruencia $t^2 \equiv N\ (mod\ p)$ tiene o bien 0 o bien 2 soluciones módulo p. Más generalmente, las congruencias

$$t^2 \equiv N\ (mod\ p^e)$$

módulo potencias de p tienen o bien 0 o bien 2 soluciones. Esto hace que tamizar potencias de primos impares sea relativamente directo. Tamizar con potencias de 2 es un poco más engañoso, ya que el número de soluciones puede ser diferente módulo 2, módulo 4, y módulo potencias más altas de 2. Además, puede haber más de 2 soluciones. Por ejemplo, $t^2 \equiv N\ (mod\ 8)$ tiene cuatro soluciones diferentes módulo 8 si $N \equiv 1\ (mod\ 8)$. Así que, si bien tamizar potencias de 2 no es intrínsecamene difícil, debe tratarse como un caso especial.

4.7.3. El tamiz de cuerpo numérico

El tamiz de cuerpo numérico es un método de factorización que trabaja en un anillo más grande que el de los enteros. Los detalles completos escapan a este trabajo, pero en esta sección explicaremos brevemente las ideas que hacen que este método sea el más rápido conocido para factorizar números grandes de la forma $N = pq$, con p y q primos de orden aproximadamente igual. Para factorizar N, empezaremos buscando un entero m no nulo y un polinomio mónico irreducible $f(x) \in \mathbb{Z}[x]$ de grado bajo satisfaciendo

$$f(m) \equiv 0\ (mod\ N).$$

Sea d el grado de $f(x)$ y sea β una raíz de $f(x)$ (posiblemente compleja). Trabajaremos en el anillo

$$\mathbb{Z}[\beta] = \{c_0 + c_1\beta + \cdots + c_{d-1}\beta^{d-1} \in \mathbb{C} : c_0, c_1, \cdots, c_{d-1} \in \mathbb{Z}\}.$$

Notemos que aunque escribimos $\mathbb{Z}[\beta]$ como un subanillo de los complejos, no es necesario trabajar con números reales o complejos. Podemos trabajar con $\mathbb{Z}[\beta]$ en forma puramente algebraica ya que es el cociente $\mathbb{Z}[x]/(f(x))$.

El siguiente paso es hallar un número grande de pares de enteros $(a_1, b_1), \cdots, (a_k, b_k)$ que satisfagan simultaneamente

$$\prod_{i=1}^{k}(a_1 - b_i m) = A^2 \quad y \quad \prod_{i=1}^{k}(a_1 - b_i\beta) = \alpha^2.$$

Por definición de $\mathbb{Z}[\beta]$, podemos encontrar una expresión para α de la forma

$$\alpha = c_0 + c_1\beta + \cdots + c_{d-1}\beta^{d-1} \quad con \quad c_0, c_1, \cdots, c_{d-1} \in \mathbb{Z}.$$

Recordemos que asumimos al comienzo que $f(m) \equiv 0 \ (mod\ N)$. Lo que significa que

$$m \equiv \beta \ (mod\ N) \quad \text{en el anillo } \mathbb{Z}[\beta].$$

Luego por un lado tenemos que

$$A^2 \equiv \alpha^2 \ (mod\ N) \ \text{ en el anillo } \mathbb{Z}[\beta].$$

Y por el otro lado tenemos que

$$\alpha \equiv c_o + c_1 m + \cdots + c_{d-1}m^{d-1} \ (mod\ N) \text{ en el anillo } \mathbb{Z}[\beta].$$

Así

$$A^2 \equiv (c_o + c_1 m + \cdots + c_{d-1}m^{d-1})^2 \ (mod\ N).$$

Por lo que hemos creado una congruencia $A^2 \equiv B^2 \ (mod\ N)$ válida en $\mathbb{Z}$, y como es usual, hay una buena chance de que $\text{MCD}(A - B, N)$ produzca un factor no trivial de N. ¿Cómo hallamos los pares (a_i, b_i) que necesitamos? Para la primer productoria, podemos usar algún algoritmo de tipo tamiz, similar al método usado en el tamiz cuadrático., para hallar valores de $a - bm$ que sean suaves, y usar algebra lineal para hallar un subconjunto con las propiedades deseadas. La idea de Pollard es, simultáneamente hacer algo similar con la segunda productoria, trabajando en $\mathbb{Z}[\beta]$. Buscando así, pares de enteros (a, b) tales que $a - b\beta$ es "suave" en $\mathbb{Z}[\beta]$. Surgen varias cuestiones, potenciales dificultades, cuando uno trata de hacer esto. Los detalles como dijimos antes, escapan a las necesidades de este trabajo.

Sin embargo, vamos a comentar más sobre el primer paso del algoritmo. Primero necesitamos un entero m y un polinomio mónico irreducible $f(x)$ de grado bajo talque $f(m) \equiv 0 \ (mod\ N)$. El truco es primero elegir el grado deseado de f, luego elegir un entero m talque

$$(N/2)^{1/d} < m < N^{1/d},$$

y luego escribir N en la base m,

$$N = c_0 + c_1 m + \cdots + c_{d-1}m^{d-1} + c_d m^d \ \ con\ 0 \leq c_i < m.$$

La condición sobre m asegura que $c_d = 1$, asi podemos tomar f mónico

$$f(x) = c_0 + c_1 x + \cdots + c_{d-1}x^{d-1} + x^d.$$

Necesitamos además que $f(x)$ sea irreducible, pero si $f(x)$ se factorizase en $\mathbb{Z}[x]$, digamos $f(x) = g(x)h(x)$, luego $N = f(m) = g(m)h(m)$ nos da una factorización de N y listo. Ahora que tenemos m y $f(x)$ podemos iniciar el algoritmo. No hay duda de que este tamiz es más complicado que el cuadrático, pero su mayor utilidad radica en que la cantidad de números que tiene que considerar es mucho menor.

4.8. El método de cálculo de índice para calcular logaritmos discretos en $\mathbb{F}_p$

El cálculo de índice es un método para resolver el PLD en un cuerpo finito $\mathbb{F}_p$. El método usa números suaves y tiene cierta similaridad a los tamices estudiados previamente, por lo cual lo estudiamos aquí y no en el capítulo anterior. La idea es simple, recordemos que queremos resolver el PLD

$$g^x \equiv h \ (mod\ p),$$

con p primo, y los enteros g y h son datos. Para simplicidad, asumimos que g es raíz primitiva modulo p, así sus potencias generan $\mathbb{F}_p^*$. Más que resolver el problema original directamente, en su lugar elegimos un valor B y resolvemos el PLD

$$g^x \equiv \ell \ (mod\ p) \ \text{ para todos los primos } \ell \leq B.$$

En otras palabras, calculamos el logaritmo discreto $log_g(\ell)$ para cada primo $\ell \leq B$. Una vez hecho esto, buscamos en las cantidades

$$h \cdot g^{-k} \ (mod\ p) \ \text{ para } k = 1, 2, \ldots$$

hasta que encontramos un valor de k tal que $h \cdot g^{-k}$ $(mod\ p)$ sea B-suave. Para este valor de k tenemos que

$$h \cdot g^{-k} \equiv \prod_{\ell \leq B} \ell^{e_\ell} \ \ (mod\ p)$$

para ciertos exponentes e_ℓ. Reescribimos la igualdad anterior en términos de logaritmos discretos como

$$log_g(h) \equiv k + \sum_{\ell \leq B} e_\ell \cdot log_g(\ell) \ \ (mod\ p-1),$$

donde, recordemos, los logaritmos discretos están definidos sólo módulo $p-1$. Pero hemos asumido que calculamos $log_g(\ell)$ para todos los primos $\ell \leq B$. Así la última igualdad nos dá una formula para $log_g(h)$. Nos falta explicar como hallar $log_g(\ell)$ para primos pequeños ℓ. La idea, nuevamente, es simple: Seleccionamos al azar exponentes i y calculamos

$$g_i \equiv gi \ (mod\ p) \quad \text{con } 0 < g_i < p.$$

Si g_i no es B-suave, lo descartamos. Si g_i es B-suave, podemos factorizarlo como

$$g_i = \prod_{\ell \leq B} \ell^{u_\ell}(i).$$

En términos de logaritmos discreto, esto nos da la relación

$$i \equiv log_g(g_i) \equiv \sum_{\ell \leq B} u_\ell(i) \cdot log_g(\ell) \ \ (mod\ p-1).$$

Notemos que aquí las únicas cantidades desconocidas en la formula son los logaritmos discretos $log_g(\ell)$. Así que si encontramos más de $\pi(B)$ ecuaciones como la anterior, podremos usando algebra lineal hallar todos esos logaritmos. Estos los logaritmos son los"índices" que le dan el nombre al método.

Observación 4.8.1. Una pequeña dificultad que hemos ignorado es el hecho de que las ecuaciones lineales son congruencias modulo $p-1$. Los métodos de malgebra lineal estándares como la eliminación gaussiana no funcionan bien modulo números compuestos, pues, hay números que no tienen inversos multiplicativos. El TCR resuelve esto. primero resolovemos las congruencias modulo q para cada q dividiendo $p-1$. Luego si q aparece en la factorización de $p-1$ a una potencia q^e, llevamos la solución de $\mathbb{Z}/q\mathbb{Z}$ a $\mathbb{Z}/q^e\mathbb{Z}$. Finalmente, usamos el TCR para combinar soluciones modulo potencias de primos para obtener una solución modulo $p-1$. En las aplicaciones criptográficas uno debería elegir p tal que $p-1$ sea divisible por un primo grande, en otro caso, el algoritmo de Pohlig-Hellman que hemos visto, resuelve el PLD.

Los detalles del tiempo necesario y algunos otros inconvenientes no los veremos en este trabajo.

4.9. Encriptación probabilística y el criptosistema Goldwasser-Micali

Supongamos que Ana quiere usar un CCP ara encriptar y mandar a Bruno un bit, es decir Ana quiere mandar a Bruno uno de los valores 0 o 1. A primera vista, tal arreglo parece inherentemente inseguro. Todo lo que Inés tiene que hacer es encriptar los dos posibles mensajes $m = 0$ y $m = 1$, y luego comparar las encriptaciones con el texto cifrado de Ana. Más generalmente, en cualquier criptosistema para el cual el conjunto de posibles mensajes es pequeño, Inés puede encriptar cualquier mensaje usando la clave pública de Bruno hasta que encuentre el que es el de Ana. La encriptación probabilística fue inventada por Goldwasser y Micali como una solución a este problema. La idea es que Ana elija el mensaje m y una secuencia aleatroia de información r, y luego usa la clave pública de Bruno para encriptar el par (m, r). Idealmente, como r varía sobre todos sus posibles valores, los texto cifrados para (m, r) van a variar .ªleatoriamente"sobre todos los posibles textos cifrados. Más precisamente para cualquier m_1 y m_2 fijados, y para r variando, la distribución de valores de las dos cantidades

$$\begin{aligned} e(m_1, r) &= \text{ el texto cifrado para el mensaje } m_1 \text{ y la secuencia aleatroia } r, \\ e(m_2, r) &= \text{ el texto cifrado para el mensaje } m_1 \text{ y la secuencia aleatroia } r, \end{aligned}$$

deberían ser esencialemente indistinguibles. Notar que no es necesario que Bruno sea capaz de recuperar el par (m, r) por completo mientras realiza la decriptación. Él necesita recuperar solamente el mensaje m. Esta idea abstracta es clara, pero ¿cómo creamos un esquema de encriptación probabilística en la práctica? Goldwasser y Micali describen uno, el cual aunque es impracticable (ya que encripta sólo un bit por vez) tiene la ventaja de ser muy fácil de describir y analizar. La idea esta basada en la dificultad del siguiente problema:

Sean p y q primos (secretos) y sea $N = pq$ dado. Para un entero dado a, determinar si a es un cuadrado módulo N, es decir determinar si existe un entero u satisfaciendo que $u^2 \equiv a \ (mod\ N)$.

Notar que Bruno, quien sabe cómo factorizar $N = pq$, es capaz de resolver este problema muy fácilmente, ya que

$$a \text{ es un cuadrado módulo } pq \quad \text{si y sólo si} \quad \left(\frac{a}{p}\right) = 1 \text{ y } \left(\frac{a}{q}\right) = 1.$$

Inés, por otro lado, tiene más trabajo, ya que ella sólo conoce el valor de N. Ella puede calcular $\left(\frac{a}{N}\right)$, pero como vimos antes esto no le dice a ella si a es un cuadrado módulo N. Goldwasser y Micali explotaron este hecho para crear un CCP probabilístico que describiremos a continuación.

- **Creación de clave**: Brueno elige primos secretos p y q, y elige a tal que $\left(\frac{a}{p}\right) = \left(\frac{a}{q}\right) = -1$. Luego publica $N = pq$ y a.
- **Encriptación**: Ana elige un mensaje $m \in 0, 1$, elige al azar r con $1 < r < N$, y usa la clave pública de Bruno (N, a) para calcular

$$c = \begin{cases} r^2 \ (mod\ N) & \text{si } m = 0, \\ ar^2 \ (mod\ N) & \text{si } m = 1. \end{cases}$$

  Por último envía el texto cifrado c a Bruno
- **Decriptación**: Bruno calcula $\left(\frac{c}{p}\right)$. Decripta así

$$m = \begin{cases} 0 & \text{si } \left(\frac{c}{p}\right) = 1, \\ 1 & \text{si } \left(\frac{c}{p}\right) = -1. \end{cases}$$

Es fácil ver que este sistema funciona pues

$$\left(\frac{c}{p}\right) = \begin{cases} \left(\frac{r^2}{p}\right) = \left(\frac{r}{p}\right)^2 = 1 & \text{si } m = 0, \\ \left(\frac{ar^2}{p}\right) = \left(\frac{a}{p}\right)\left(\frac{r}{p}\right)^2 = \left(\frac{a}{p}\right) = -1 & \text{si } m = 1. \end{cases}$$

Más aún, como Ana elige r al azar, el conjunto de valores que Inés ve cuando Ana encripta $m = 0$ consiste en todos los posibles cuadrados módulo N, y el conjunto de valores que Inés ve cuando Ana encripta $m = 1$ consiste en todos los posibles números c tales que $\left(\frac{c}{N}\right) = 1$ que no son cuadrados módulo N. ¿Qué información obtiene Inés si computa el símbolo de Jacobi $\left(\frac{c}{N}\right)$ (que puede hacerlo pues N es público)? Si $m = 0$, luego $c \equiv r^2 \ (mod\ N)$, así

$$\left(\frac{c}{N}\right) = \left(\frac{r^2}{N}\right) = \left(\frac{r}{N}\right)^2 = 1.$$

Por otro lado, si $m = 1$, luego $c \equiv ar^2 \ (mod\ N)$, así

$$\left(\frac{c}{N}\right) = \left(\frac{ar^2}{N}\right) = \left(\frac{a}{N}\right) = \left(\frac{a}{pq}\right) = \left(\frac{a}{p}\right)\left(\frac{a}{q}\right) = (-1) \cdot (-1) = 1$$

que también es igual a 1. De esta manera, $\left(\frac{c}{N}\right)$ es siempre 1, sin importar el valor de N, y en definitiva, el símbolo de Jacobi no le da a Inés información útil.

Observación 4.9.1. El criptosistema Goldwasser Micali es impracticable porque cada bit del mensaje es encriptado con un número módulo N. Para que sea seguro es necesario que Inés sea incapaz de factorizar el número $N = pq$ así que en la práctica, N debe ser (al menos) un número de 1000 bits. Luego, si Ana quiere mandar un mensaje a Bruno de k bits, su texto cifrado será de longitud $1000k$ bits. Luego, este criptosistema tiene un radio de expansión de mensaje de 1000, ya que el texto cifrado es 1000 veces más grande que el mensaje original.

Referencias

[1] D.S.Dummit y R.M.Foote, *Abstract Algebra*. John Wiley & Sons Inc., Hoboken, NJ, tercera edición, 2004.

[2] T.W. Hungerford, *Algebra*. Springer Science, octava edición, 2003.