UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES

A C A T L Á N

CRIPTOGRAFÍA DE CLAVE PÚBLICA CON CURVAS ELÍPTICAS SOBRE CAMPOS FINITOS

TESINA

QUE PARA OBTENER EL TÍTULO DE

LICENCIADO EN MATEMÁTICAS APLICADAS Y COMPUTACIÓN

PRESENTA:

LUIS ALONSO LÓPEZ GARCÍA

ASESOR: DR. FRANCISCO MARCOS LÓPEZ GARCÍA

NOVIEMBRE DE 2004

Agradecimientos

A mi madre, porque eres lo más hermoso que tengo en la vida. Te dedico este trabajo como una expresión de agradecimiento y del inmenso amor que te tengo.

A mi hermano Marcos, quien fue fundamental para que haya realizado este trabajo. Gracias por orientar mi camino hacia la consolidación de este trabajo y por estar siempre pendiente de mí.

A mis hermanas Sagrario, Alejandra que a pesar de la distancia sé que están conmigo. Les agradezco la comprensión, el cariño y el apoyo incondicional a mis proyectos.

A mi hermana Hortencia (Dora), por su enorme apoyo moral y por soportar mis malos momentos.

A los sinodales que leyeron este trabajo y me hicieron llegar oportunamente sus comentarios y sugerencias que contribuyeron sustancialmente en la presentación de este trabajo.

A la Universidad Nacional Autónoma de México por acogerme en su seno. Espero responder con creces todo el apoyo recibido.

A mis amigos y amigas Ángela, Aquilino, Ignacio, Josué, Luis Felipe, Nancy, Oscar y Pavel por su leal y verdadera amistad.

Criptografía de clave pública con curvas elípticas sobre campos finitos

Luis Alonso López García

24 de noviembre de 2004

Índice general

Introducción

Durante los últimos años hemos presenciado la gran importancia que ha llegado a adquirir el manejo y la transmisión electrónica de la información, la cual presenta muchas ventajas, entre ellas, la capacidad de transmisión instantánea, acceso remoto, operaciones financieras y comerciales. Sin embargo, a diferencia de la documentación impresa, los datos electrónicos que viajan y se almacenan por medios electrónicos pueden ser robados, alterados y manipulados desde una localidad distante. Por lo anterior se han ideado diversos mecanismos para proteger el contenido de la información mientras ésta se almacena ó viaja hacia su destino, una de estas técnicas es la criptografía.

La criptografía es el estudio de técnicas matemáticas para enviar mensajes de manera disfrazada a través de un canal de comunicación inseguro, como lo puede ser Internet. Un criptosistema mapea unidades de texto ordinario llamadas unidades de texto claro en unidades de texto codificado llamadas unidades de texto cifrado. La seguridad que provee un criptosistema, se basa en el hecho de que al tratar de romper el cifrado se debe resolver un problema matemático de gran complejidad; es decir, que la solución a dicho problema tomará un tiempo muy grande en términos prácticos.

Entre los servicios prácticos más importantes que están relacionados directamente a la criptografía podemos mencionar: cifrado de datos, firma digital, dinero digital, elecciones digitales, autenticación de usuarios, protocolo de conocimiento nulo, distribución de claves y tarjetas inteligentes (smart cards).

El desarrollo más notable en la historia de la criptografía se dió en 1976 cuando dos investigadores de la universidad de Stanford, Whitfield Diffie y Martin Hellman, publican el artículo "*New Directions in Cryptography*" [8], en donde introducen el concepto de *función tramposa* y con ello la *idea de la criptografía de clave pública.* Una función tramposa es una función fácil de evaluar pero difícil de invertir, al menos que se tenga información adicional. Además idearon un método para que dos entidades lograran compartir información secreta sin tener una comunicación previa. Dicho método se basa en el problema de resolver logaritmos discretos en campos finitos, conocido como el problema del logaritmo discreto.

Dos años más tarde **R**ivest, **S**hamir y **A**dleman [1] propusieron una implementación de una función tramposa, dando origen al conocido criptosistema **RSA**. La seguridad del **RSA** se basa en el problema de la factorización de un número compuesto grande n.

Mas tarde en 1984, Taher ElGamal [10] propuso los criptosistemas de clave pública basados en el problema del logaritmo discreto. El criptosistema ElGamal se ha refinado e incorporado en la firma digital **DSS**.

Así lucía el panorama de la criptografía de clave pública hasta 1985, cuando Neal Koblitz [21] y Víctor Miller [30] propusieron un criptosistema de clave pública análogo al esquema del ElGamal en la cual el grupo $\mathbb{F}_q^*$ es reemplazado por el grupo de puntos sobre una curva elíptica definida sobre un campo finito $\mathbb{F}_q$.

La seguridad de los criptosistemas de curvas elípticas se basa en la dificultad del problema del logaritmo discreto en curvas elípticas y en los últimos diez años se ha intensificado su estudio tratando de probar o desaprobar la seguridad de los criptosistemas de curvas elípticas.

Hoy por hoy, se considera que la criptografía con curvas elípticas es el criptosistema más eficiente de la criptografía moderna, con niveles de seguridad similares a los ofrecidos por otros criptosistema tales como el **RSA**, ElGamal.

En este trabajo revisamos algunos conceptos matemáticos básicos relacionados con la criptografía con curvas elípticas.

Capítulo 1

Fundamentos matemáticos

En este capítulo presentamos varios conceptos matemáticos y resultados que se utilizarán en el desarrollo posterior de este trabajo. La exposición se enmarca en tres aspectos fundamentales: el primero hace referencia a la teoría de números, el segundo trata de la herramienta matemática necesaria para comprender a las curvas elípticas sobre campos finitos, a saber, grupos, anillos y campos; la parte final se dedica a describir algunas transformaciones afines que utilizamos en el Capítulo 2.

A lo largo de este trabajo $\mathbb{N}$ denotará el conjunto de los números naturales, $\mathbb{Z}$ el conjunto de los números enteros, $\mathbb{Q}$ el conjunto de los números racionales, $\mathbb{R}$ los números reales y $\mathbb{C}$ los números complejos. Por último $\mathbb{Z}^+ = \{z \in \mathbb{Z} : z > 0\}$ denota el conjunto de enteros positivos.

El n-ésimo producto cartesiano de un conjunto $K \neq \phi$ está dado por

$$K^n = K \times \ldots \times K = \{(x_1, \ldots, x_n) : x_i \in K,\ i = 1, \ldots, n\}.$$

1.1. Teoría de números

1.1.1. Divisibilidad y el algoritmo extendido de Euclides

Definición 1.1.1 *Dados a, $b \in \mathbb{Z}$, decimos que a divide a b si existe un entero d tal que $b = ad$ y lo denotamos por $a \mid b$. En este caso decimos que a es un* divisor *de b.*

Todo entero $b > 1$ tiene al menos dos divisores positivos: 1 y b.

Definición 1.1.2 *Dado $p \in \mathbb{Z}^+$, $p > 1$ es un número primo si y sólo si sus únicos divisores positivos son 1 y p.*

Si α es un entero no negativo entonces usamos la notación $p^\alpha \parallel b$ para denotar que p^α es la potencia más grande de p que divide a b, es decir, que $p^\alpha \mid b$ y que $p^{\alpha+1} \nmid b$.

El *teorema fundamental de la aritmética* establece que cualquier número natural n puede escribirse de manera única (excepto por el orden de los factores) como el producto de un número finito de números primos, explícitamente:

$$n = p_1^{\alpha_1} p_2^{\alpha_2} \dots p_k^{\alpha_k} = \prod_{i=1}^{k} p_i^{\alpha_i}. \tag{1.1}$$

En donde $p_1, p_2, \dots, p_k$ son números primos con $p_1 < p_2 < \dots < p_k$ y cada $\alpha_i \in \mathbb{Z}^+$ (ver [24, pag. 171]).

Por ejemplo, el número 49014212940000 lo podemos escribir como $2^5 3^{10} 5^4 7^3 11^2$.

Una consecuencia del teorema fundamental de la aritmética es que proporciona un método sistemático para encontrar todos los divisores positivos de n, una vez que n está escrito como un producto de potencias de primos. Cualquier divisor d de n debe ser un producto de los mismos primos pero no excediendo la potencia que divide exactamente a n . Esto es, si $p^\alpha \parallel n$ entonces $p^\beta | d$ siempre que $0 \leq \beta \leq \alpha$. Por ejemplo, ya que $120 = 2^3 3^1 5^1$, un divisor positivo de 120 debe ser de la forma $d = 2^{\alpha_1} 3^{\alpha_2} 5^{\alpha_3}$, en donde α_1 puede tomar los valores 0, 1, 2, 3; α_2 y α_3 pueden ser 0 ó 1.

En general, el entero $n = p_1^{\alpha_1} p_2^{\alpha_2} \dots p_k^{\alpha_k}$ tiene $(\alpha_1 + 1)(\alpha_2 + 1) \cdots (\alpha_k + 1)$ divisores positivos (ver [17, pag. 12]).

Un entero positivo n es *libre de cuadrados* si tiene la representación $n = p_1 p_2 \dots p_k$, por ejemplo $105 = 3 \cdot 5 \cdot 7$ es libre de cuadrados.

Definición 1.1.3 *El máximo común divisor de dos enteros positivos a, b, con $ab \neq 0$, es el entero más grande que divide a a y b; tal entero lo denotamos por $m.c.d.(a, b)$.*

Definición 1.1.4 *Decimos que $a, b \in \mathbb{Z}^+$ son primos relativos si $m.c.d.(a, b) = 1$, es decir, si a y b no tienen divisores comunes mayores que 1. Lo anterior se suele denotar por $(a, b) = 1$.*

Si conocemos la factorización de a y b entonces es fácil calcular $m.c.d.(a, b)$. A saber, elegimos todos los primos que aparecen en ambas factorizaciones elevados a la mínima potencia de los exponentes respectivos. Por ejemplo, al comparar la factorización de $10780 = 2^2 \cdot 5 \cdot 7^2 \cdot 11$ con la factorización de $4200 = 2^3 \cdot 3 \cdot 5^2 \cdot 7$ vemos que $m.c.d.(4200, 10780) = 2^2 \cdot 5 \cdot 7 = 140$.

Proposición 1.1.5 *Sea $d = m.c.d.(a, b)$ con $a > b > 0$. Entonces existen enteros u y v tal que $d = au + bv$. En otras palabras, el máximo común divisor de dos números puede expresarse como una combinación lineal de los números con coeficientes enteros.*

El algoritmo extendido de Euclídes

En primer lugar, enunciamos un resultado aparentemente trivial pero muy importante en la teoría de números, a saber, el *teorema de la división en* $\mathbb{Z}$.

Teorema 1.1.6 *Si* $a \in \mathbb{Z}$ *y* $b \in \mathbb{Z}^+$ *entonces existen enteros únicos* q *y* r *tales que*

$$a = bq + r \quad y \quad 0 \leq r < b$$

(ver [24, pag. 67]).

Decimos que r es el mínimo residuo no negativo que se obtiene de dividir a a entre b.

El algoritmo de Euclídes se basa en el siguiente teorema.

Teorema 1.1.7 *Sean* $a, b \in \mathbb{Z}^+$ *y* r *el residuo cuando* a *es dividido por* b. *Entonces* $m.c.d.(a,b) = m.c.d.(b,r)$ *(ver [24, pag. 161]).*

Ahora describimos el algoritmo de Euclídes en su forma básica, el cual es un método eficiente para calcular $m.c.d.(a,b)$ aún cuando no conozcamos los factores primos de a y b.

Sean a, $b \in \mathbb{Z}^+$ tal que $a \geq b$. Si $a = b$ entonces $m.c.d.(a,b) = a$, así que asumimos que $a > b$. (Si esto no se cumple entonces los intercambiamos). Con la aplicación sucesiva del teorema de la división, obtenemos las siguientes igualdades:

$$\begin{aligned}
a &= q_1 b + r_1, \quad 0 \leq r_1 < b \\
b &= q_2 r_1 + r_2, \quad 0 \leq r_2 < r_1 \\
&\vdots \\
r_{n-2} &= q_n r_{n-1} + r_n, \quad 0 \leq r_n < r_{n-1} \\
r_{n-1} &= q_{n+1} r_n + 0.
\end{aligned}$$

Dado que los residuos son no negativos y cada vez más pequeños entonces esta secuencia debe terminar en algún momento, cuando el residuo r_{n+1} es igual a cero.

Por inducción se sigue que $m.c.d.(a,b) = m.c.d.(b,r_1) = m.c.d.(r_1,r_2) = \ldots = m.c.d.(r_{n-1},r_n) = r_n$ (el último residuo no cero).

En el siguiente ejemplo vemos como funciona el algoritmo de Euclídes.

Ejemplo 1.1.8 *Con la aplicación sucesiva del algoritmo de la división podemos calcular* $m.c.d.(11200, 3533)$:

$$\begin{aligned} 11200 &= 3 * 3533 + 601 \\ 3533 &= 5 * 601 + 528 \\ 601 &= 1 * 528 + 73 \\ 528 &= 7 * 73 + 17 \\ 73 &= 4 * 17 + 5 \\ 17 &= 3 * 5 + 2 \\ 5 &= 2 * 2 + 1 \\ 2 &= 2 * 1 + 0. \end{aligned}$$

Por lo tanto $m.c.d.(11200, 3533) = 1$.

La información que proporciona el Algoritmo de Euclídes no siempre es suficiente para muchos problemas.

Por la Proposición 1.1.5 sabemos que $d = m.c.d.(a, b)$, puede expresarse como $d = au + bv$ con $u, v \in \mathbb{Z}$. Para poder calcular los valores de u y v frecuentemente es necesario extender el algoritmo de Euclídes. Una forma de hacer esto se muestra con el siguiente ejemplo.

Ejemplo 1.1.9 *Usando las igualdades del Ejemplo 1.1.8 en orden inverso, podemos escribir* $m.c.d.(11200, 3533)$ *como una combinación lineal de 11200 y 3533 como sigue,*

$$\begin{aligned} 1 &= 5 - 2 * 2 \\ &= 5 * 7 - 2 * 17 \\ &= 73 * 7 - 17 * 30 \\ &= 73 * 217 - 30 * 528 \\ &= 601 * 217 - 528 * 247 \\ &= 601 * 1452 - 3533 * 247 \\ &= 11200 * 1452 - 3533 * 4603 \\ &= 11200 * 1452 + 3533 * (-4603). \end{aligned}$$

En [6, pag. 16-19] se describen algunas variantes a este algoritmo para aumentar su eficacia al momento de implementarlo.

Definición 1.1.10 *Sea* $n \in \mathbb{Z}^+$. *La función de Euler* $\varphi(n)$ *denota el número de primos relativos a* n.

Algunas de las propiedades de la función de Euler son:

1. Si p es primo entonces $\varphi(p) = p - 1$.
2. La función de Euler es multiplicativa. Esto es, si $(m, n) = 1$ entonces $\varphi(mn) = \varphi(m)\varphi(n)$.
3. Si p es primo entonces $\varphi(p^\alpha) = p^\alpha - p^{\alpha-1} = p^\alpha(1 - \frac{1}{p})$.
4. Si $n = p_1^{\alpha_1} p_2^{\alpha_2} \dots p_k^{\alpha_k}$ es la factorización canónica de n, de 2. y 3. se tiene que

$$\varphi(n) = p_1^{\alpha_1}\left(1 - \frac{1}{p_1}\right) p_2^{\alpha_2}\left(1 - \frac{1}{p_2}\right) \dots p_k^{\alpha_k}\left(1 - \frac{1}{p_k}\right) = n \prod_{p|n}\left(1 - \frac{1}{p}\right). \quad (1.2)$$

Como una consecuencia de la fórmula para $\varphi(n)$ tenemos el siguiente resultado, el cual usaremos cuando discutamos el criptosistema RSA en el Capítulo 3.

Proposición 1.1.11 *Supongamos que se sabe que n es el producto de dos números primos distintos. Entonces el conocimiento de esos dos primos p y q es equivalente al conocimiento de $\varphi(n)$.*

La prueba se puede consultar en [17, pag. 22].

1.1.2. Congruencias

En el año de 1801, Carl F. Gauss introdujo en su trabajo *Disquisitiones Arithmeticae* una nueva notación relacionada con la noción de divisibilidad, la cual ha resultado fructífera.

Definición 1.1.12 *Dos enteros a y b son congruentes módulo n si $n \mid (a - b)$ y se denota*

$$a \equiv b \ mod \ n.$$

El conjunto de enteros que son congruentes a a módulo n es:

$$\{a + nk \mid k \in \mathbb{Z}\}.$$

El siguiente resultado se obtiene de la definición anterior e interpreta a la congruencia en términos de igualdades entre números.

Proposición 1.1.13 *$a \equiv b \ mod \ n$ si y sólo si $a = b + kn$ para algún entero k.*

Del Teorema 1.1.6 y la proposición anterior tenemos,

Proposición 1.1.14 *Sea $n \in \mathbb{N}$, $n > 1$. Todo entero a es congruente módulo n a exactamente uno de los números en el conjunto $\{0, 1, \ldots, n-1\}$.*

El teorema 1.1.6 también nos proporciona un criterio para determinar cuando dos números enteros son congruentes módulo n :

Proposición 1.1.15 *$a \equiv b \ mod\ n$ si y sólo si a y b tienen el mismo mínimo residuo no negativo cuando se dividen por n.*

Definición 1.1.16 *Si $n \in \mathbb{N}$, $a \in \mathbb{Z}$ y $a \equiv r \ mod\ n$, donde $0 \leq r < n$, entonces decimos que r es el mínimo residuo no negativo de a módulo n.*

Ahora presentamos las propiedades básicas de las congruencias (ver [24, pag. 216]).

Proposición 1.1.17 *Sean $n \in \mathbb{N}, n > 1$ y $a, b, c, a', b', k \in \mathbb{Z}$.*

1. *Si $a \equiv b \ mod\ n$ entonces $ka \equiv kb \ mod\ n$;*
2. *Si $a \equiv b \ mod\ n$ y $b \equiv c \ mod\ n$ entonces $a \equiv c \ mod\ n$;*
3. *Si $a \equiv b \ mod\ n$ y $a' \equiv b' \ mod\ n$ entonces:*
   a) $a + a' \equiv b + b' mod\ n$ y
   b) $aa' \equiv bb' mod\ n$.

Recordamos que una *relación de equivalencia* en un conjunto no vacío S, es una relación binaria que es reflexiva (xRx para todo x en S), simétrica (si xRy entonces yRx) y transitiva (si xRy y yRz entonces xRz). La *clase de equivalencia* de $a \in S$ es por definición el subconjunto $[a] = \{x \in S : x \sim a\}$, donde $\sim$ denota la relación de equivalencia.

La congruencia módulo n es una relación de equivalencia en el conjunto $\mathbb{Z}$. En consecuencia la congruencia módulo n particiona al conjunto $\mathbb{Z}$ en clases de equivalencia llamadas *clases de congruencia.*

La clase de congruencia de un entero a es

$$[a] = \{a + sn \mid s \in \mathbb{Z}\}.$$

Denotamos al conjunto de las clases de congruencia por

$$\mathbb{Z}_n = \{[0], [1], \ldots, [n-1]\}.$$

Dado que todo entero a es congruente a un único entero r que satisface $0 \leq r < n$, la clase $[a]$ contiene un único entero r tal que $0 \leq r < n$; de esta manera se tiene que hay una correspondencia inyectiva entre $\mathbb{Z}_n$ y $\{0, 1, \ldots, n-1\}$ y se suelen identificar los dos conjuntos.

Ahora enunciamos el *pequeño Teorema de Fermat* (ver [24, pag. 317]) y algunas de sus consecuencias.

Teorema 1.1.18 *Si $a \in \mathbb{Z}^+$ y p es un primo con $(a,p) = 1$, entonces $p \mid (a^{p-1} - 1)$. Esto es, $a^{p-1} \equiv 1 \ mod\ p$.*

Corolario 1.1.19 *Si $a \in \mathbb{Z}^+$, entonces $a^p \equiv a \ mod\ p$ para cualquier primo p.*

Corolario 1.1.20 *Si $r \equiv s \ mod\ p-1$ entonces $a^r \equiv a^s \ mod\ p$.*

El siguiente Teorema, debido a Euler, es una generalización del Teorema 1.1.18.

Teorema 1.1.21 *Si $a, n \in \mathbb{Z}^+$ con $(a,n) = 1$, entonces $a^{\varphi(n)} \equiv 1 \ mod\ n$ (ver [24, pag. 332]).*

Corolario 1.1.22 *Si $(a,m) = 1$ y si n' es el mínimo residuo no negativo de n módulo $\varphi(m)$, entonces $a^n \equiv a^{n'} \ mod\ m$.*

1.1.3. Exponenciación modular utilizando el método de cuadrados repetidos

Un cálculo común que se presenta en la aritmética modular es encontrar $b^n \bmod m$ (es decir, encontrar el mínimo residuo no negativo) cuando m y n son muy grandes. Hay una manera de hacerlo que es mucho más fácil que la multiplicación repetida de b por sí misma. En lo subsecuente suponemos que $b < m$ y que después de multiplicar reducimos inmediatamente módulo m (es decir, reemplazamos al producto por su mínimo residuo no negativo). De esta manera nunca nos encontraremos con enteros mayores que m^2. A continuación describimos el algoritmo.

Usamos a para denotar el producto parcial. Empezamos con $a = 1$. Los dígitos binarios de n los denotamos por $n_0, n_1, \ldots, n_{k-1}$, es decir,

$$n = n_0 + 2n_1 + 4n_2 + \ldots + 2^{k-1}n_{k-1}.$$

Cada n_j es 0 ó 1. Si $n_0 = 1$, cambiar a por b (de lo contrario mantener $a = 1$). Entonces elevamos al cuadrado a b, y poner $b_1 = b^2 \bmod m$ (es decir, b_1 es el mínimo residuo no negativo de $b^2 \bmod m$). Si $n_1 = 1$ multiplicar a por b_1 (y reducir *mod m*); de lo contrario a no cambia. A continuación elevamos al cuadrado a b_1, y ponemos $b_2 = b_1^2 \bmod m$. Si $n_2 = 1$, multiplicar a por b_2; de lo contrario a no cambia. Continuando de esta forma, podemos observar que en el j–ésimo paso habremos calculado $b_j \equiv b^{2^j} \bmod m$. Si $n_j = 1$, es decir si 2^j aparece en la expansión binaria de n, entonces se incluye b_j en el producto por a (si 2^j está ausente en n entonces no). Después del paso $(k-1)$ tendremos $a \equiv b^n \bmod n$.

Ejemplo 1.1.23 *Calcular* 3^{55} *mod* 25. *Primero notamos que* $55 = (110111)_2$, $a = 1$ *y* $b = 3$. *Ahora calculamos los mínimos residuos no negativos de* 3^2 *y sus sucesivos cuadrados módulo* 25*:*

$$\begin{array}{lll} n_0 = 1 & a = 3 & b_1 \equiv 3^2 \equiv 9 \ mod\ 25, \\ n_1 = 1 & a = 2 & b_2 \equiv 3^4 \equiv 9^2 \equiv 6 \ mod\ 25, \\ n_2 = 1 & a = 12 & b_3 \equiv 3^8 \equiv 6^2 \equiv 11 \ mod\ 25, \\ n_3 = 0 & a = 12 & b_4 \equiv 3^{16} \equiv 11^2 \equiv 21 \ mod\ 25, \\ n_4 = 1 & a = 2 & b_5 \equiv 3^{32} \equiv 21^2 \equiv 16 \ mod\ 25, \\ n_5 = 1 & a = 7. & \end{array}$$

De esta manera tenemos que $7 \equiv 3^{55}$ *mod* 25.

1.2. Grupos, anillos y campos finitos

El orden que se seguirá para esta sección es ascendente con respecto a la riqueza de las estructuras algebraicas que se presentan, se inicia con la definición y las propiedades básicas de los grupos, se continúa con los anillos y finalmente con los campos.

1.2.1. Grupos

Definición 1.2.1 *Un grupo* $\langle \mathcal{G}, * \rangle$ *es un conjunto no vacío* $\mathcal{G}$ *sobre el cual se ha definido una operación binaria* $*$ *que satisface:*

1. $x * y \in \mathcal{G}$ *para todo* $x, y \in \mathcal{G}$. *Propiedad de cerradura.*
2. $x * (y* z) = (x * y) * z$ *para todo* $x, y, z \in \mathcal{G}$. *Propiedad asociativa.*
3. *Existe* $e \in \mathcal{G}$ *tal que* $x * e = e * x = x$ *para todo* $x \in \mathcal{G}$. *Propiedad de la existencia del elemento neutro.*
4. *Para cada* $x \in \mathcal{G}$ *existe* $x' \in \mathcal{G}$ *tal que* $x * x' = x' * x = e$. *Propiedad de la existencia de inversos.*

Si además de cumplir con las propiedades anteriores se satisface que $x * y = y * x$ para todo $x, y \in \mathcal{G}$ (conmutatividad), entonces se dice que $\langle \mathcal{G}, * \rangle$ es un *grupo abeliano.*

Por simplicidad en algunos casos denotaremos a un grupo abeliano $\langle \mathcal{G}, * \rangle$ por $\mathcal{G}$.

Ejemplo 1.2.2 *En* $\mathbb{Z}_n$ *definimos una operación* $+$ *dada por* $[a] + [b] = [a + b]$; *es decir, para sumar* $[a], [b] \in \mathbb{Z}_n$ *elegimos representantes* $a \in [a]$, $b \in [b]$ *de cada clase, sumamos en* $\mathbb{Z}$ *estos representantes y luego consideramos la clase de equivalencia de* $a + b$. *De la Proposición 1.1.17 se sigue que esta operación está bien definida y que* $(\mathbb{Z}_n, +)$ *es un grupo abeliano.*

Para introducir otro ejemplo presentamos el siguiente resultado.

Proposición 1.2.3 *Si $a \in \mathbb{Z}_n$ y $(a,n) = 1$ entonces existe un único $b \in \mathbb{Z}_n$ tal que $ab \equiv 1$ mod n, y denotamos $b = a^{-1}$.*

Ahora consideramos el conjunto $\mathbb{Z}_n^* = \{a \in \mathbb{Z}_n : (a,n) = 1\}$. En $\mathbb{Z}_n^*$ definimos una operación $\cdot$ dada por $[a] \cdot [b] = [a \cdot b]$. De las Proposiciones 1.1.17 y 1.2.3, se puede ver que esta operación está bien definida y que $(\mathbb{Z}_n^*, \cdot)$ es un grupo abeliano llamado el grupo multiplicativo de $\mathbb{Z}_n$.

En particular, si n es un número primo p entonces

$$\mathbb{Z}_p^* = \{a \in \mathbb{Z}_p : 1 \leq a \leq p-1\}.$$

Comentario 1.2.4 *El algoritmo extendido de Euclídes (Sección 1.1.1) nos permite calcular los inversos de elementos en el grupo multiplicativo $(\mathbb{Z}_n^*, \cdot)$. Por ejemplo, para calcular el inverso de* 11200 *mod* 3533 *usamos del Ejemplo 1.1.9 que* $1 = 11200 * 1452 + 3533 * (-4603)$, *por lo tanto* $11200 * 1452 \equiv 1$ *mod* 3533 *y así* $(11200)^{-1} = 1452$ *en* $\mathbb{Z}_{3533}^*$.

Comentario 1.2.5 *Si conocemos la factorización canónica de n entonces podemos calcular $\phi(n)$ utilizando la Ecuación (1.2). Del Teorema 1.1.21 se tiene que $a^{-1} = a^{\phi(n)-1}$ en $\mathbb{Z}_n^*$.*

Una propiedad importante de un grupo $\mathcal{G}$ es el número de elementos con que cuenta. A este número se le conoce como el orden del grupo y se le denota por $|\mathcal{G}|$. Cuando el grupo tiene orden finito se dice que el grupo es finito. Por ejemplo $|\mathbb{Z}_n^*| = \varphi(n)$.

Es frecuente encontrar en la literatura sobre grupos la notación con el uso de exponentes, por ejemplo $a^m \in \mathcal{G}, m \geq 1$ representa al elemento $a * \cdots * a$ (m factores) y $a^{-m} = (a^{-1})^m$.

Teorema 1.2.6 *Sea $\mathcal{G}$ un grupo abeliano finito. Entonces para cualquier $a \in \mathcal{G}$, $a^{|\mathcal{G}|} = e$. (ver [5, pag. 149])*

Definición 1.2.7 *El orden de un elemento $a \in \langle \mathcal{G}, * \rangle$ es el entero positivo más pequeño m tal que $a^m = e$ y se denota por $m = ord(a)$.*

Definición 1.2.8 *Un grupo $\mathcal{G}$ es cíclico si existe $a \in \mathcal{G}$ tal que para cada $b \in \mathcal{G}$ existe un entero m tal que $b = a^m$. Al elemento a se le llama un generador o elemento primitivo del grupo cíclico $\mathcal{G}$.*

Cuando el grupo es finito tenemos una definición equivalente a la anterior,

Definición 1.2.9 *Un grupo finito $\mathcal{G}$ es cíclico si existe $a \in \mathcal{G}$ con $ord(a) = |\mathcal{G}|$. Se dice que a es un generador de $\mathcal{G}$.*

Definición 1.2.10 *Un subconjunto no vacío $\mathcal{H}$ de un grupo $\mathcal{G}$ se dice que es un subgrupo de $\mathcal{G}$ si respecto a la operación definida en $\mathcal{G}$, él mismo forma un grupo.*

Teorema 1.2.11 *Sea $\mathcal{G}$ un grupo y $a \in \mathcal{G}$, entonces*

$$\mathcal{H} = \{a^n \mid n \in \mathbb{Z}\}$$

es un subgrupo cíclico de $\mathcal{G}$ (ver [12, pag. 57]).

Teorema 1.2.12 *Todo grupo cíclico es abeliano (ver [12, pag. 57]).*

Teorema 1.2.13 *Todo subgrupo de un grupo cíclico es cíclico (ver [12, pag. 58]).*

A continuación damos el teorema básico respecto a los generadores de subgrupos para los grupos cíclicos finitos.

Teorema 1.2.14 *Sea $\mathcal{G}$ un grupo cíclico finito y a un generador. Si $b = a^s$ entonces b genera un subgrupo cíclico $\mathcal{H}$ de $\mathcal{G}$ con $|\mathcal{G}|/d$ elementos donde $d = m.c.d.(|\mathcal{G}|, s)$.*

Corolario 1.2.15 *Si a es un generador de un grupo cíclico finito $\mathcal{G}$, entonces a^r es un generador de $\mathcal{G}$ si y sólo si $(r, |\mathcal{G}|) = 1$.*

Ejemplo 1.2.16 *Tenemos que 2 es un generador de $\mathbb{Z}_{11}^*$, ya que al calcular 2^i mod 11, $i = 1, 2, \ldots, 10$ se obtienen todos los elementos de $\mathbb{Z}_{11}^*$.*

Ahora se enuncia un resultado importante en la teoría de grupos, a saber, el teorema de Lagrange.

Teorema 1.2.17 *Si $\mathcal{G}$ es un grupo finito y $\mathcal{H}$ es un subgrupo de $\mathcal{G}$ entonces $|\mathcal{H}|$ divide a $|\mathcal{G}|$. (Ver [12, pag. 112])*

Como consecuencia de lo anterior se tiene que,

Corolario 1.2.18 *Si $\mathcal{G}$ es un grupo finito con $|\mathcal{G}| = p$, p primo, entonces $\mathcal{G}$ es un grupo cíclico. De hecho, cualquier $x \in \mathcal{G}$, $x \neq e$ es un generador de $\mathcal{G}$.*

1.2.2. Anillos y campos

A diferencia de los grupos, en los anillos se definen dos operaciones, a las cuales comúnmente se les identifica con la suma y el producto.

Definición 1.2.19 *Un anillo $\langle \mathcal{R}, +, * \rangle$ es un conjunto no vacío $\mathcal{R}$ junto con dos operaciones (denotadas por $+$ y $*$) tal que:*

1. *$\langle \mathcal{R}, + \rangle$ es un grupo abeliano.*

2. $x * (y * z) = (x * y) * z$ *para todo* $x, y, z \in \mathcal{R}$. *Propiedad asociativa del producto.*

3. $x * (y + z) = (x * y) + (x * z)$ *y* $(x + y) * z = (x * z) + (y * z)$ *para todo* $x, y, z \in \mathcal{R}$. *Propiedad distributiva.*

Es posible que exista un elemento $1 \in \mathcal{R}$ con la propiedad de que $1 * x = x = x * 1$ para todo $x \in \mathcal{R}$, si tal elemento existe se dice que $\mathcal{R}$ es un *anillo con uno*. Por otro lado, si el producto en $\mathcal{R}$ tiene la propiedad de que $x * y = y * x$ para todo $x, y \in \mathcal{R}$ entonces se dice que $\mathcal{R}$ es un *anillo conmutativo*.

Ejemplo 1.2.20 *En* $(\mathbb{Z}_n, +)$ *definimos una operación producto* $*$ *dada por* $[a]*[b] = [a * b]$ *para todo* $[a], [b] \in \mathbb{Z}_n$. *De las propiedades de congruencia se sigue que* $(\mathbb{Z}_n, +, *)$ *es un anillo conmutativo con uno.*

Definición 1.2.21 *Sea* $\mathcal{R}$ *un anillo con uno. Un elemento* $u \in \mathcal{R}$, $u \neq 0$ *se le llama unidad de* $\mathcal{R}$ *si existe* $v \in \mathcal{R}$ *tal que* $uv = 1 = vu$. *Al elemento* $v \in \mathcal{R}$ *se le llama inverso multiplicativo de* u.

En un anillo con uno $\mathcal{R}$, el conjunto de unidades de $\mathcal{R}$ (denotado $\mathcal{R}^*$) forma un grupo con la multiplicación de $\mathcal{R}$.

Ahora veamos como relacionar los anillos mediante funciones.

Definición 1.2.22 *Sean* $\mathcal{A}$ *y* $\mathcal{B}$ *dos anillos. Un morfismo (homomorfismo) de anillos* $f : \mathcal{A} \to \mathcal{B}$ *es una función tal que:*

1. $f(a + b) = f(a) + f(b)$,

2. $f(a * b) = f(a) * f(b)$ *para todo* $a, b \in \mathcal{A}$.

Debemos notar que la suma $(+)$ y el producto $(*)$ en el lado izquierdo de estas igualdades son las operaciones en $\mathcal{A}$ y en el lado derecho son las operaciones en $\mathcal{B}$.

Lema 1.2.23 *Si* $f : \mathcal{A} \to \mathcal{B}$ *es un morfismo de anillos entonces*

1. $f(0) = 0$

2. $f(-a) = -f(a)$ *para todo* $a \in \mathcal{A}$.

Definición 1.2.24 *Un morfismo de anillos* $f : \mathcal{A} \to \mathcal{B}$ *es un isomorfismo si es biyectivo.*

Definición 1.2.25 *Un campo* $\mathbb{K} = \langle \mathbb{K}, +, * \rangle$ *es un conjunto no vacío* $\mathbb{K}$ *junto con dos operaciones* $+$ *y* $*$, *que contiene dos elementos distintos* $0, 1$ *y que cumple:*

1. $\langle \mathbb{K}, +, 0 \rangle = \mathbb{K}^+$ *y* $\langle \mathbb{K} - \{0\}, *, 1 \rangle = \mathbb{K}^*$ *son grupos abelianos.*

2. $x*(y+z) = (y+z)*x = (x*y)+(x*z)$ *para todo* $x, y, z \in \mathbb{K}$. *Propiedad distributiva.*

De esta manera, un campo es un anillo conmutativo tal que todo elemento diferente de cero tiene un inverso multiplicativo.

De manera semejante a nuestra notación en teoría de grupos, nos referiremos de manera algo incorrecta, a un campo $\mathbb{K}$ en lugar de $\langle \mathbb{K}, +, * \rangle$.

Ejemplo 1.2.26 *Los conjuntos $\mathbb{Q}$, $\mathbb{R}$ y $\mathbb{C}$ (con las operaciones aritméticas usuales) son campos.*

Al igual que los grupos, el *orden* de un campo es el número de elementos que contiene y se denota por $|\mathbb{K}|$.

Definición 1.2.27 *Un campo $\mathbb{K}$ es finito si $|\mathbb{K}| < \infty$.*

Proposición 1.2.28 *Sea $\mathbb{K}$ un campo finito, entonces el grupo multiplicativo $\mathbb{K}^*$ es cíclico (ver [5, pag. 354]).*

Teorema 1.2.29 *$\mathbb{Z}_p$ es un campo si y sólo si p es primo.*

Ejemplo 1.2.30 *Si p es primo entonces el grupo multiplicativo $\mathbb{Z}_p^*$, con la identidad $[1]$, es cíclico.*

Definición 1.2.31 *Para un primo p, sea $\mathbb{F}_p = \{0, 1, 2, \ldots, p-1\}$ y $\psi : \mathbb{Z}_p \longmapsto \mathbb{F}_p$ la función dada por*

$$\psi([a]) = a \text{ para } a = 0, 1, 2, \ldots, p-1.$$

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

Llevar a cabo cálculos con los elementos de $\mathbb{F}_p$ significa realizar operaciones aritméticas de enteros módulo p.

Definición 1.2.32 *Sea $\mathbb{K}$ un campo. Si existe un entero n tal que $\overbrace{a + \ldots + a}^{n-veces} = 0$ para cada $a \in \mathbb{K}$, $a \neq 0$ entonces al mínimo entero positivo k que satisface lo anterior se le llama la característica del campo. Si no existe tal entero k, se dice que la característica del campo es 0.*

Teorema 1.2.33 *Si $\mathbb{K}$ es un campo de característica p, entonces p es primo. (ver [12, pag. 263])*

Teorema 1.2.34 *Un campo finito $\mathbb{F}$ tiene por característica un número primo (ver [25]).*

Anillo de polinomios

Si $\mathbb{K}$ es un campo entonces $\mathbb{K}[x]$ denota al conjunto de polinomios en la indeterminada x con coeficientes en $\mathbb{K}$. Los elementos de $\mathbb{K}[x]$ son expresiones formales de la forma

$$f(x) = \sum_{i=0}^{m} a_i x^i = a_0 + a_1 x + \ldots + a_m x^m, \text{ con } a_i \in \mathbb{K}.$$

Dos polinomios $f(x) = a_0 + a_1 x + \ldots + a_m x^m$ y $g(x) = b_0 + b_1 x + \ldots + b_n x^n$ son iguales si y sólo si los coeficientes de cada potencia de x son iguales: $a_0 = b_0$, $a_1 = b_1, \ldots, a_n = b_n, \ldots, a_m = b_m$. En particular, si $n < m$, entonces $b_{n+1} = b_{n+2} = \ldots = b_m = 0$.

Definición 1.2.35 *Sea $\mathbb{K}$ un campo. Para $f(x) \in \mathbb{K}[x]$ definimos el grado de f, denotado $gra(f)$, como el más grande de todos los exponentes en los términos de f con coeficientes no nulos.*

El polinomio con $a_0 = a_1 = \cdots = 0$ es el *polinomio cero* y se denota por 0. Por convención, el polinomio cero tiene grado -1. Cualquier otro polinomio $f(x) \in \mathbb{K}[x]$ tiene grado mayor o igual que cero.

En $\mathbb{K}[x]$ se definen las operaciones usuales de suma y producto de polinomios, explícitamente:

Ejemplo 1.2.36 *1. Si $f(x) = a_0 + a_1 x + \ldots + a_m x^m$ y $g(x) = b_0 + b_1 x + \ldots + b_n x^n$ son dos polinomios en $\mathbb{K}[x]$ con $m \geq n$ se define*

$$(f+g)(x) = (a_0 + b_0) + (a_1 + b_1)x + \ldots + (a_n + b_n)x^n + a_{n+1}x^{n+1} + \ldots + a_m x^m.$$

2. Si $f(x), g(x) \in \mathbb{K}[x]$ son dos polinomios como antes, se define

$$(f \cdot g)(x) = a_0 b_0 + (a_0 b_1 + a_1 b_0)x + \ldots + \left(\sum_{i+j=k} a_i b_j \right) x^k + \ldots + a_m b_n x^{m+n}.$$

Con las operaciones definidas anteriormente, el polinomio cero y el polinomio 1 dado por $a_0 = 1$ y $a_1, \ldots, a_m = 0$, es fácil verificar que $(\mathbb{K}[x], +, \cdot)$ es un anillo conmutativo con uno.

El algoritmo extendido de Euclídes para $\mathbb{K}[x]$

Enunciamos a continuación el teorema de la división para $\mathbb{K}[x]$.

Teorema 1.2.37 *Sea $\mathbb{K}$ un campo. Si $f(x), g(x) \in \mathbb{K}[x]$ con $f(x) \neq 0$, entonces existen polinomios $q(x), r(x) \in \mathbb{K}[x]$ con $gra(r) < gra(f)$ tal que*

$$g(x) = f(x)q(x) + r(x);$$

además, $q(x)$ y $r(x)$ son únicos (ver [5, pag. 240]).

Definición 1.2.38 *Sean* $f(x), g(x) \in \mathbb{K}[x]$. *Decimos que* $f(x)$ divide *a* $g(x)$ *si* $g(x) = f(x)q(x)$ *para algún* $q(x) \in \mathbb{K}[x]$ *y se denota por* $f(x)|g(x)$.

Definición 1.2.39 *Sean* $\mathbb{K}$ *un campo y* $f(x), g(x) \in \mathbb{K}[x]$. *Un polinomio* $p(x) \in \mathbb{K}[x]$ *es un máximo común divisor de* $f(x)$ *y* $g(x)$ *(denotado por* $m.c.d.(f, g)$*) si cumple:*

1. $p(x)|f(x)$ *y* $p(x)|g(x)$,
2. *cualquier* $q(x) \in \mathbb{K}[x]$ *que divide a* $f(x)$ *y a* $g(x)$ *tiene un grado el cual no es más grande que el grado de* $p(x)$.

Podemos encontrar "un" $m.c.d.(f, g)$ usando el teorema de la división repetidamente, así como lo hicimos con los números naturales (sección 1.1.1). A este proceso se le conoce como el *algoritmo de Euclídes para polinomios.* Veamos como funciona:

$$\begin{aligned}
g(x) &= f(x)q_1(x) + r_1(x),\ 0 \leq gra\ (r_1) < gra\ (f) \\
f(x) &= r_1(x)q_2(x) + r_2(x),\ 0 \leq gra\ (r_2) < gra\ (r_1) \\
r_1(x) &= r_2(x)q_3(x) + r_3(x),\ 0 \leq gra\ (r_3) < gra\ (r_2) \\
&\vdots \\
r_{n-2}(x) &= r_{n-1}(x)q_n(x) + r_n(x),\ 0 \leq gra\ (r_n) < gra\ (r_{n-1}) \\
r_{n-1}(x) &= r_n(x)q_{n+1}(x) + 0.
\end{aligned}$$

Ya que $gra(r_1) < gra(f), gra(r_2) < gra(r_1), \ldots$ la secuencia de divisiones termina después de a lo más $gra(f)$ pasos. Tenemos que $r_n = m.c.d.(f, g)$ (el último residuo no cero).

Dos polinomios pueden tener varios máximos comunes divisores, de hecho, el algoritmo de Euclídes puede producir más de uno de ellos.

Tenemos un resultado para polinomios que es análogo a la Proposición 1.1.5.

Proposición 1.2.40 *Sean* $f(x), g(x) \in \mathbb{K}[x]$ *y* $d(x) = m.c.d.(f, g)$. *Entonces existen polinomios* $r(x), s(x) \in \mathbb{K}[x]$ *tal que* $d(x) = r(x)f(x) + s(x)g(x)$.

El algoritmo extendido de Euclídes para polinomios nos proporciona un método para obtener de manera explícita los polinomios $r(x)$ y $s(x)$.

Si $\mathbb{K}$ es un campo, entonces en el anillo $\mathbb{K}[x]$ las únicas unidades son las constantes no cero, esto es, los polinomios de grado cero.

Definición 1.2.41 *Un polinomio no constante* $p(x) \in \mathbb{K}[x]$ *es irreducible en* $\mathbb{K}[x]$ *si siempre que* $p(x) = f(x)g(x)$ *con* $f(x), g(x) \in \mathbb{K}[x]$, *entonces* $f(x)$ *ó* $g(x)$ *debe ser una unidad en* $\mathbb{K}[x]$, *esto es, un polinomio constante no nulo.*

Los polinomios irreducibles en $\mathbb{K}[x]$ cumplen un papel análogo al de los número primos. En particular,

Proposición 1.2.42 *Si* $p(x)$ *es irreducible en* $\mathbb{K}[x]$ *y si* $f(x) \in \mathbb{K}[x]$ *no es divisible por* $p(x)$ *entonces* $m.c.d.(p, f) = 1$ *(ver [5, pag. 249]).*

Congruencias módulo un polinomio $m(x)$

Definición 1.2.43 *Sean $\mathbb{K}$ un campo y $m(x) \in \mathbb{K}[x]$ con $gra\,(m(x)) \geq 1$. Dos polinomios $f(x), g(x) \in \mathbb{K}[x]$ son congruentes módulo $m(x)$, que se denota por*

$$f(x) \equiv g(x) \ mod\ m(x),$$

si $m(x)|\,(f(x) - g(x))$ ó de manera equivalente si $f(x) = g(x) + m(x)q(x)$ para algún $q(x) \in \mathbb{K}[x]$.

La congruencia módulo $m(x)$ tiene propiedades similares a las descritas en la Proposición 1.1.17.

Definición 1.2.44 *La clase de congruencia de $g(x)$ módulo $m(x)$, escrita como $[g(x)]$, es el conjunto de polinomios $f(x) \in \mathbb{K}[x]$ que son congruentes a $g(x)$ módulo $m(x)$. De esta manera*

$$\begin{aligned}[g(x)] &= \{f(x) \in \mathbb{K}[x] : f(x) \equiv g(x) \ mod\ m(x)\} \\ &= \{f(x) \in \mathbb{K}[x] : f(x) = g(x) + m(x)q(x) \text{ para algún } q(x) \in \mathbb{K}[x]\}.\end{aligned}$$

Al conjunto de clases de congruencias de polinomios en $\mathbb{K}[x]$ módulo $m(x)$ se le denota generalmente por $\mathbb{K}[x]/m(x)$.

Si $m(x)$ tiene grado d entonces por el Teorema 1.2.37 cualquier polinomio $g(x) \in \mathbb{K}[x]$ puede ser dividido por $m(x)$:

$$g(x) = m(x)q(x) + r(x),$$

en donde el residuo $r(x)$ tiene grado menor que d y es el único polinomio con esa propiedad, entonces $[g(x)] = [r(x)]$. De esta manera, toda clase de congruencia módulo $m(x)$ se representa por un único polinomio $r(x)$ de grado menor que $gra(m\,(x))$. Por lo tanto, el conjunto de polinomios $r(x)$ de grado menor que $gra(m)$ es un conjunto representativo para $\mathbb{K}[x]/m(x)$. Esta propiedad es análoga a la de que los números $0, 1, \ldots, n-1$ forman un conjunto representativo para $\mathbb{Z}_n$.

Ejemplo 1.2.45 *Sea $m(x) = x^3 + x + 1 \in \mathbb{F}_2[x]$. Del Teorema 1.2.37 se sigue que existen $q(x), r(x) \in \mathbb{F}_2[x]$ tal que $gra(r\,(x)) < 3$ y que satisfacen:*

$$g(x) = (x^3 + x + 1)q(x) + r(x),$$

y así

$$g(x) \equiv r(x) \ mod\ (x^3 + x + 1).$$

Entonces cualquier $g(x) \in \mathbb{F}_2[x]$ es congruente módulo $m(x)$ a un polinomio de grado menor que 3. Por ejemplo, $x^4 = (x^3+x+1)x+(x^2+x)$ y así $x^4 \equiv x^2+x \ mod\ (x^3+x+1)$.

Como en los enteros, definimos la suma y la multiplicación de clases de congruencias por:

$$\begin{aligned}[f(x)] + [g(x)] &= [f(x) + g(x)], \\ [f(x)] \cdot [g(x)] &= [f(x) \cdot g(x)],\end{aligned}$$

estas operaciones junto con el hecho de que

$$[-f(x)] = -[f(x)],\ [0] = 0 \text{ y } [1] = 1,$$

hacen que $\mathbb{K}[x]/m(x)$ sea un anillo conmutativo con uno.

Proposición 1.2.46 *Sea $\mathbb{K}$ un campo y $m(x) \in \mathbb{K}[x]$ irreducible en $\mathbb{K}[x]$ con $gra\,(m\,(x)) \geq 1$, entonces $\mathbb{K}[x]/m(x)$ es un campo (ver [5, pag. 417]).*

La proposición anterior nos proporciona una manera de construir nuevos campos. A saber, si empezamos con un campo $\mathbb{K}$ y buscamos un polinomio $m(x) \in \mathbb{K}[x]$ irreducible en $\mathbb{K}[x]$ con $gra\,(m(x)) \geq 1$ entonces $\mathbb{F}[x]/m(x)$ es un campo.

Ejemplo 1.2.47 *Sea $\mathbb{K} = \mathbb{F}_p$ y $m(x) \in \mathbb{K}[x]$ irreducible en $\mathbb{K}[x]$ con grado $d \geq 1$, entonces $\mathbb{F}[x]/m(x)$ tiene p^d elementos. Más adelante explicaremos como representar a los p^d elementos de $\mathbb{F}[x]/m(x)$.*

Definición 1.2.48 *Sean $\mathbb{K}$ un campo y $f(x) = a_0 + a_1x + \ldots + a_nx^n \in \mathbb{K}[x]$. Dado $b \in \mathbb{K}$, al hecho de reemplazar a x por b para obtener $f(b) = a_0 + a_1b + \ldots + a_nb^n \in \mathbb{K}$ se le llama principio de sustitución.*

Definición 1.2.49 *A $b \in \mathbb{K}$ se le llama cero (ó raíz) del polinomio $f(x) \in \mathbb{K}[x]$ si $f(b) = 0$.*

Una conexión importante entre las raíces y la divisibilidad está dada por el siguiente teorema.

Teorema 1.2.50 *$b \in \mathbb{K}$ es un cero del polinomio $f(x) \in \mathbb{K}[x]$ si y sólo si $(x - b)|f(x)$ (ver [25, pag. 27]).*

Teorema 1.2.51 *Un polinomio $f(x) \in \mathbb{K}[x]$ de grado 2 ó 3 es irreducible en $\mathbb{K}[x]$ si y sólo si $f(x)$ no tiene ceros en $\mathbb{K}$ (ver [25, pag. 28]).*

Extensión de campos

Definición 1.2.52 *Si $\mathbb{K}$ y $\mathbb{L}$ son campos tal que $\mathbb{K} \subseteq \mathbb{L}$ y las operaciones definidas en $\mathbb{K}$ son las mismas que las de $\mathbb{L}$ entonces decimos que $\mathbb{K}$ es un subcampo de $\mathbb{L}$ ó que $\mathbb{L}$ es una extensión de $\mathbb{K}$. Usamos la notacion $\mathbb{L}/\mathbb{K}$ para denotar que $\mathbb{L}$ es una extensión de $\mathbb{K}$. Además, si $\mathbb{K} \neq \mathbb{L}$ decimos que $\mathbb{K}$ es un subcampo propio de $\mathbb{L}$.*

Definición 1.2.53 *A un campo que no contiene subcampos propios se le llama campo primo.*

Ejemplo 1.2.54 *Si $\mathbb{K}$ es un subcampo de $\mathbb{F}_p$, p primo, entonces $\mathbb{K}$ debe contener los elementos $0, 1$ y por lo tanto a todos los demás elementos de $\mathbb{F}_p$ (por la propiedad de cerradura de $\mathbb{K}$ bajo la suma) y así $\mathbb{K} = \mathbb{F}_p$. Se sigue que $\mathbb{F}_p$ es un campo primo.*

Otro ejemplo de campo primo es el conjunto de números racionales $\mathbb{Q}$.

Por otro lado, la intersección de cualquier colección no vacía de subcampos de un campo dado $\mathbb{K}$ es un subcampo de $\mathbb{K}$. Si tomamos la intersección de todos los subcampos de $\mathbb{K}$, obtenemos el llamado *subcampo primo* de $\mathbb{K}$ que es obviamente un campo primo.

Teorema 1.2.55 *Un campo $\mathbb{K}$, o es de característica p y contiene un subcampo isomorfo a $\mathbb{F}_p$ o es de característica 0 y contiene un subcampo isomorfo a $\mathbb{Q}$ (ver [12, pag. 263]).*

Definición 1.2.56 *Sea $\mathbb{L}/\mathbb{K}$ una extensión de campos y $M \subseteq \mathbb{L}$. El campo $\mathbb{K}(M)$ se define como la intersección de todos los subcampos de $\mathbb{L}$ que contienen a M y $\mathbb{K}$, se le llama la extensión de $\mathbb{K}$ obtenida al adjuntar los elementos de M. Para el conjunto finito $M = \{\theta_1, \ldots, \theta_n\}$ escribimos $\mathbb{K}(M) = \mathbb{K}(\theta_1, \ldots, \theta_n)$. Si M consiste de un sólo elemento $\theta \in \mathbb{L}$ entonces se dice que $E = \mathbb{K}(\theta)$ es una extensión simple de $\mathbb{K}$ y a θ se le llama el elemento definido de $\mathbb{L}$ sobre $\mathbb{K}$.*

Ahora definimos un tipo de extensión importante.

Definición 1.2.57 *Sea $\mathbb{L}/\mathbb{K}$ una extensión de campos. Un elemento $\theta \in \mathbb{L}$ es algebraico sobre $\mathbb{K}$, si existe un polinomio $p(x) \in \mathbb{K}[x]$ no cero, tal que $p(\theta) = 0$. Una extensión $\mathbb{L}/\mathbb{K}$ se llama algebraica sobre $\mathbb{K}$ (ó es una extensión algebraica sobre $\mathbb{K}$) si todos los elementos de $\mathbb{L}$ son algebraicos sobre $\mathbb{K}$.*

Si $\mathbb{L}/\mathbb{K}$ es una extensión de campos entonces es claro que cualquier elemento $a \in \mathbb{K} \subseteq \mathbb{L}$ es algebraico sobre $\mathbb{K}$ ya que es el cero del polinomio $x - a \in \mathbb{K}[x]$. Así que es más interesante buscar elementos algebraicos en $\mathbb{L} - \mathbb{K}$.

Ejemplo 1.2.58 *En la extensión $\mathbb{R}/\mathbb{Q}$ el elemento $\sqrt{2} \in \mathbb{R} - \mathbb{Q}$ es algebraico sobre $\mathbb{Q}$ ya que es un cero del polinomio $x^2 - 2 \in \mathbb{Q}[x]$. Así, existen elementos algebraicos no racionales.*

Si $\mathbb{L}/\mathbb{K}$ es una extensión de campos entonces $\mathbb{L}$ puede considerarse como un *espacio vectorial sobre* $\mathbb{K}$. Esto es porque los elementos de $\mathbb{L}$ (vectores) forman primero que nada, un grupo abeliano bajo la suma. Además, todo vector $\alpha \in \mathbb{L}$ puede multiplicarse por un escalar $r \in \mathbb{K}$, así que $r\alpha \in \mathbb{L}$ y además se cumple: $r(\alpha+\beta) = r\alpha+r\beta$, $(r+s)\alpha = r\alpha+s\alpha$, $(rs)\alpha = r(s\alpha)$, $1\alpha = \alpha$ donde r, $s \in \mathbb{K}$ y α, $\beta \in \mathbb{L}$.

Definición 1.2.59 *Sea $\mathbb{L}/\mathbb{K}$ una extensión de campos. Si $\mathbb{L}$, considerado como espacio vectorial sobre $\mathbb{K}$, tiene dimensión finita entonces se dice que $\mathbb{L}$ es una extensión finita de $\mathbb{K}$. La dimensión del espacio vectorial $\mathbb{L}$ sobre $\mathbb{K}$ es el grado de $\mathbb{L}$ sobre $\mathbb{K}$, en símbolos $[\mathbb{L} : \mathbb{K}]$.*

Teorema 1.2.60 *Toda extensión finita de $\mathbb{K}$ es algebraica sobre $\mathbb{K}$.*

Teorema 1.2.61 *Dada $\mathbb{L}/\mathbb{K}$ una extensión de campos y $\theta \in \mathbb{L}$ algebraico sobre $\mathbb{K}$, la extensión simple $\mathbb{K}(\theta)$ es una extensión finita de $\mathbb{K}$ y por lo tanto algebraica (ver [25]).*

Definición 1.2.62 *Sea $\mathbb{K}$ un campo. Decimos que $f(x) \in \mathbb{K}[x]$ es un polinomio mónico si el coeficiente de su termino con el exponente más grande es 1.*

Proposición 1.2.63 *Sea $\mathbb{L}/\mathbb{K}$ una extensión de campos y $\theta \in \mathbb{L}$ algebraico sobre $\mathbb{K}$. El polinomio mónico de menor grado $m(x) \in \mathbb{K}[x]$ del cual $\theta \in \mathbb{L}$ es raíz, es único. Más aún, $m(x)$ es irreducible en $\mathbb{K}[x]$ y $m(x)$ divide a cualquier otro polinomio $p(x) \in \mathbb{K}[x]$ del cual θ es raíz.*

Definición 1.2.64 *Al polinomio mónico irreducible $m(x) \in \mathbb{K}[x]$ de grado menor del cual $\theta \in \mathbb{L}$ es raíz, se le llama polinomio mínimo de θ sobre $\mathbb{K}$.*

Teorema 1.2.65 *Sea $\theta \in \mathbb{L}$ un elemento algebraico sobre $\mathbb{K}$ de grado n y sea g el polinomio mínimo de θ sobre $\mathbb{K}$. Entonces:*

1. *$\mathbb{K}(\theta)$ es isomorfo a $\mathbb{K}[x]/g$*
2. *$[\mathbb{K}(\theta) : \mathbb{K}] = n$ y $\{1, \theta, \ldots, \theta^{n-1}\}$ es una base de $\mathbb{K}(\theta)$ sobre $\mathbb{K}$.*

Debe señalarse que el Teorema anterior funciona bajo la suposición de que tanto $\mathbb{K}$ y θ están incluídos en un campo. Esto es necesario para que las expresiones algebraicas que involucran a θ tengan sentido. La prueba puede consultarse en [25, pag. 33]. Se tiene entonces que los elementos de una extensión algebraica simple $\mathbb{K}(\theta)$ de $\mathbb{K}$ son polinomios en θ. De hecho, cualquier elemento en $\mathbb{K}(\theta)$ puede ser representado de forma única como:

$$a_0 + a_1\theta + \ldots + a_{n-1}\theta^{n-1} \text{ con } a_i \in \mathbb{K} \text{ para } 0 \leq i \leq n-1.$$

Por ahora sólo queremos construir una extensión algebraica simple sin hacer alusión a un campo que lo contenga.

Teorema 1.2.66 *Sea $f(x) \in \mathbb{K}[x]$ irreducible en $\mathbb{K}[x]$. Entonces existe una extensión algebraica simple $\mathbb{L}$ de $\mathbb{K}$ con un cero $\theta \in \mathbb{L}$ de $f(x)$ como un elemento definido de $\mathbb{L}$ sobre $\mathbb{K}$ (ver [25, pag. 34]).*

Veamos un ejemplo utilizando este Teorema.

Ejemplo 1.2.67 *Por el Teorema 1.2.51 el polinomio $f(x) = x^2 + x + 2 \in \mathbb{F}_3[x]$ es irreducible en $\mathbb{F}_3[x]$. El teorema previo implica que existe una extensión algebraica simple $\mathbb{L}$ de $\mathbb{F}_3$ que contiene algún cero $\theta \in \mathbb{L}$ de $f(x)$. Por el Teorema 1.2.65(2.) ó por la estructura conocida del campo $F_3[x]/f(x)$, la extensión simple $\mathbb{L} = \mathbb{F}_3(\theta)$ consiste de nueve elementos* 0, 1, 2, θ, $\theta + 1$, $\theta + 2$, 2θ, $2\theta + 1$, $2\theta + 2$.

Estructura de campos finitos

Esta sección esta basada en el Capítulo 2 del libro [25].

Hemos visto que para cada primo p el conjunto $\mathbb{Z}_p$ de clases de congruencia módulo p es un campo finito con p elementos (Teorema 1.2.29), el cual se identifica con el campo de Galois $\mathbb{F}_p$ de orden p (Definición 1.2.31). Los campos finitos $\mathbb{F}_p$ juegan un papel muy importante en la teoría general de campos ya que todo campo $\mathbb{K}$ de característica p debe contener una copia isomorfa de $\mathbb{F}_p$ por el Teorema 1.2.55, y por lo tanto $\mathbb{K}$ se puede pensar como una extensión de $\mathbb{F}_p$. Esta observación, junto con el hecho de que todo campo finito tiene característica prima (Teorema 1.2.34), es fundamental para la clasificación de campos finitos. Primero establecemos una condición necesaria sobre el número de elementos de un campo finito.

Lema 1.2.68 *Si $\mathbb{F}$ es un campo finito que contiene un subcampo $\mathbb{K}$, entonces $\mathbb{F}$ tiene $|\mathbb{K}|^n$ elementos, donde $n = [\mathbb{F} : \mathbb{K}]$.*

Teorema 1.2.69 *Si $\mathbb{F}$ es un campo finito con característica p entonces $\mathbb{F}$ tiene p^n elementos, donde n es el grado de $\mathbb{F}$ sobre su subcampo primo $\mathbb{K}$.*

A partir de los campos primos $\mathbb{F}_p$ podemos construir otros campos finitos usando el método de adjunción de una raiz descrito en la sección anterior, es decir, si $f(x) \in \mathbb{F}_p[x]$ es un polinomio irreducible en $\mathbb{F}_p[x]$ de grado n, entonces al adjuntar una raíz de $f(x)$ a $\mathbb{F}_p$ obtenemos un campo finito con p^n elementos.

Sin embargo, hasta el momento no está claro si para todo $n \in \mathbb{Z}^+$ existe un polinomio irreducible en $\mathbb{F}_p[x]$ de grado n. Para hacer esto utilizamos los siguientes resultados.

Proposición 1.2.70 *Si $\mathbb{F}$ es un campo finito entonces $a^{|\mathbb{F}|} = a$ para todo $a \in \mathbb{F}$.*

Definición 1.2.71 *Sea $\mathbb{F}/\mathbb{K}$ una extensión y $f \in \mathbb{K}[x]$ no constante. Se dice que f se descompone en $\mathbb{F}$ si se puede escribir como un producto de factores lineales en $\mathbb{F}[x]$, esto es, si existen $\alpha_1, \ldots, \alpha_n \in \mathbb{F}$ tal que*

$$f(x) = a(x - \alpha_1) \cdots (x - \alpha_n).$$

Se dice que el campo $\mathbb{F}$ es un campo de descomposición de f sobre $\mathbb{K}$ si f se descompone en $\mathbb{F}$ y si $\mathbb{F} = \mathbb{K}(\alpha_1, \ldots, \alpha_n)$.

Proposición 1.2.72 *Si $\mathbb{F}$ es un campo finito con $|\mathbb{F}| = q$ y $\mathbb{K}$ es un subcampo de $\mathbb{F}$, entonces el polinomio $x^q - x \in \mathbb{K}[x]$ se factoriza en $\mathbb{F}[x]$ como*

$$x^q - x = \prod_{a \in \mathbb{F}} (x - a)$$

y $\mathbb{F}$ es un campo de descomposición de $x^q - x$ sobre $\mathbb{K}$.

Teorema 1.2.73 *(Existencia.y unicidad de campos finitos). Para cada primo p y cada $n \in \mathbb{Z}^+$ existe un campo finito con p^n elementos. Además, cualquier campo finito $\mathbb{F}_q$ con $q = p^n$ elementos, es isomorfo al campo de descomposición de $x^q - x$ sobre $\mathbb{F}_p$.*

Teorema 1.2.74 *Si $\mathbb{F}_q$ es un campo finito con $q = p^n$ elementos entonces todo subcampo de $\mathbb{F}_q$ tiene orden p^m, donde m es un divisor positivo de n. Reciprocamente, si m es un divisor positivo de n entonces hay exactamente un subcampo de $\mathbb{F}_q$ con p^m elementos.*

Definición 1.2.75 *A un generador del grupo cíclico $\mathbb{F}_q^*$ se le llama generador de $\mathbb{F}_q$.*

La existencia de elementos primitivos implica que todo campo finito se puede pensar como una extensión algebraica simple de su subcampo primo,

Teorema 1.2.76 *Si $\mathbb{F}_q, \mathbb{F}_r$ son campos finitos y $\mathbb{F}_r$ es una extensión de $\mathbb{F}_q$, entonces $\mathbb{F}_r$ es una extensión algebraica simple de $\mathbb{F}_q$ y cada generador de $\mathbb{F}_r$ puede servir como un elemento definido de $\mathbb{F}_r$ sobre $\mathbb{F}_q$.*

Corolario 1.2.77 *Para cada campo finito $\mathbb{F}_q$ y cada entero positivo n existe un polinomio irreducible en $\mathbb{F}_q[x]$ de grado n.*

Teorema 1.2.78 *Si $f \in \mathbb{F}_q[x]$ es irreducible en $\mathbb{F}_q[x]$ con grado n, entonces $f(x)$ tiene una raíz α en $\mathbb{F}_{q^n}$. Además, todos las raíces de $f(x)$ son simples y están dadas por los n distintos elementos $\alpha, \alpha^q, \alpha^{q^2}, \ldots, \alpha^{q^{n-1}}$ de $\mathbb{F}_{q^n}$.*

Ahora describimos una manera de representar los elementos de un campo finito $\mathbb{F}_q$ con $q = p^n$ elementos, donde p es la característica de $\mathbb{F}_q$.

Notamos que $\mathbb{F}_q$ es una extensión simple algebraica de $\mathbb{F}_p$ por el Teorema 1.2.76. De hecho, si $f(x)$ es un polinomio irreducible en $\mathbb{F}_p[x]$ de grado n, entonces $f(x)$ tiene un cero α en $\mathbb{F}_q$ de acuerdo al Teorema 1.2.78 y así $\mathbb{F}_q = \mathbb{F}_p(\alpha)$. Entonces por el Teorema 1.2.65, todo elemento de $\mathbb{F}_q$ se escribe de manera única como un polinomio en $\mathbb{F}_p[\alpha]$ con grado menor que n. También podemos ver a $\mathbb{F}_q$ como la clase residual $\mathbb{F}_p[x]/f(x)$.

Ejemplo 1.2.79 *Para representar a los elementos de $\mathbb{F}_9$ de esta manera, consideramos a $\mathbb{F}_9$ como una extensión simple algebraica de $\mathbb{F}_3$ de grado 2, la cual se obtiene al adjuntar una raíz α de un polinomio irreducible cuadrático en $\mathbb{F}_3[x]$, a saber $f(x) = x^2 + 1 \in \mathbb{F}_3[x]$. De esta manera $f(\alpha) = \alpha^2 + 1 = 0$ en $\mathbb{F}_9$ y los nueve elementos de $\mathbb{F}_9$ están dados de la forma $a_0 + a_1\alpha$ con $a_0, a_1 \in \mathbb{F}_3$. En detalle, $\mathbb{F}_9 = \{0, 1, 2, \alpha, 1 + \alpha, 2 + \alpha, 2\alpha, 1 + 2\alpha, 2 + 2\alpha\}$.*

1.2.3. Residuos cuadráticos

Sea p un primo impar. Tenemos interés en saber cuales elementos de $\mathbb{F}_p^*$ son cuadrados. Si $a \in \mathbb{F}^*$ es un cuadrado, es decir $b^2 = a$ para algún $b \in \mathbb{F}_p^*$, entonces a tiene exactamente dos raíces cuadradas $\pm b$. De esta manera, los cuadrados en $\mathbb{F}_p^*$ pueden encontrarse al calcular $b^2 \bmod p$ para $b = 1, 2, \ldots, (p-1)/2$ (dado que los enteros restantes hasta $p-1$ son congruentes a $-b$ para una de estas b), y precisamente la mitad de los elementos en $\mathbb{F}_p^*$ son cuadrados. Por ejemplo, los cuadrados en $\mathbb{F}_{11}$ son $1^2 = 1, 2^2 = 4, 3^2 = 9, 4^2 = 5$, y $5^2 = 3$.

Definición 1.2.80 *Sea p un primo impar y $x \in \mathbb{F}_p^*$, decimos que x es un residuo cuadrático módulo p si la congruencia $y^2 \equiv x \bmod p$ tiene una solución $y \in \mathbb{F}_p^*$, en caso contrario x es un residuo no-cuadrático módulo p.*

Por la discusión anterior se tiene que hay $(p-1)/2$ residuos cuadráticos y $(p-1)/2$ residuos no cuadráticos.

Ejemplo 1.2.81 *Los residuos cuadráticos módulo 11 son $1, 3, 4, 5$ y 9. Los residuos no-cuadráticos modulo 11 son $2, 6, 7, 8, 10$.*

Si g es un generador de $\mathbb{F}_p$, entonces cualquier elemento puede escribirse en la forma g^j. De esta manera, el cuadrado de cualquier elemento es de la forma g^j con j par. Reciprocamente, cualquier elemento de la forma g^j con j par es el cuadrado de algún elemento, a saber $\pm g^{j/2}$.

El siguiente resultado, conocido como *Criterio de Euler*, es un algoritmo polinomial determinístico para calcular los residuos cuadráticos.

Teorema 1.2.82 *Sea p un primo impar. Entonces a es un residuo cuadrático módulo p si y sólo si*

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

Definición 1.2.83 *Sea a un entero y $p > 2$ un primo. El símbolo de Legendre $\left(\frac{a}{p}\right)$ está dado por*

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

Hemos visto que $a^{(p-1)/2} \equiv 1 \bmod p$ si y sólo si a es un residuo cuadrático módulo p. Si a es un múltiplo de p, entonces es claro que $a^{(p-1)/2} \equiv 0 \bmod p$. Finalmente, si a es un residuo no-cuadrático módulo p entonces $a^{(p-1)/2} \equiv -1 \bmod p$ ya que $a^{p-1} \equiv 1 \bmod p$. De hecho, el siguiente resultado proporciona un algoritmo eficiente para evaluar el símbolo de Legendre.

Proposición 1.2.84

$$\left(\frac{a}{p}\right) \equiv a^{(p-1)/2} \bmod p.$$

La prueba se puede consultar en [17, pag. 43].

1.3. Transformaciones afines

Definición 1.3.1 *Una transformación afín sobre $\mathbb{Z}_N$ es una función $f : \mathbb{Z}_N \longrightarrow \mathbb{Z}_N$ dada por $f(x) \equiv ax + b \ mod\ N$ con $a, b \in \mathbb{Z}_N$.*

Proposición 1.3.2 *Sean $a \in \mathbb{Z}_N^*, b \in \mathbb{Z}_N$. La transformación afín $f(x) \equiv ax + b \ mod\ N$ es una función biyectiva de $\mathbb{Z}_N$ en $\mathbb{Z}_N$, además $f^{-1}(y) = x \equiv a^{-1}(y - b) \ mod\ N$.*

Prueba. Si $f(x_1) \equiv f(x_2)$; entonces $ax_1 + b \equiv ax_2 + b \bmod N$, por lo tanto $ax_1 \equiv ax_2 \bmod N$. Por la Proposición 1.2.3 existe $a^{-1} \in \mathbb{Z}_N$, multiplicando la congruencia anterior por a^{-1} tenemos que $x_1 \equiv x_2 \bmod N$, de donde se sigue que f es inyectiva.

Sea $y \in \mathbb{Z}_N$, si $f(x) \equiv y \bmod N$ entonces $ax + b \equiv y \bmod N$, por lo tanto tenemos que $x \equiv a^{-1}(y - b) \bmod N$. ■

El siguiente resultado será útil para un ejemplo posterior

Proposición 1.3.3 *Sea*

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in M_2(\mathbb{Z}_N) \ y \ D = ad - bc \in \mathbb{Z}_N \backslash \{0\}.$$

los siguientes incisos son equivalentes:

1. $(D, N) = 1$; *es decir* $D \in \mathbb{Z}_N^*$,
2. *A tiene matriz inversa* $A^{-1} = \begin{pmatrix} D^{-1}d & -D^{-1}b \\ -D^{-1}c & D^{-1}a \end{pmatrix} \in M_2(\mathbb{Z}_N)$,
3. *si* $x, y \in \mathbb{Z}_N \backslash \{0\}$ *entonces* $A \begin{pmatrix} x \\ y \end{pmatrix} \neq \begin{pmatrix} 0 \\ 0 \end{pmatrix}$,
4. $T : \mathbb{Z}_N \times \mathbb{Z}_N \longrightarrow \mathbb{Z}_N \times \mathbb{Z}_N$ *dada por* $T \begin{pmatrix} x \\ y \end{pmatrix} = A \begin{pmatrix} x \\ y \end{pmatrix}$ *es biyectiva.*

(Ver [17, pag. 70]).

Para calcular el determinante D y la matriz inversa A^{-1} se opera como si estuvieramos en $\mathbb{R}$, con la variante de que las entradas se realizan módulo N.

Capítulo 2

Conceptos básicos de criptografía

El uso de la criptografía hoy en día resulta indispensable en diversos ámbitos del quehacer humano; hasta hace un par de décadas su uso estaba reservado a aspectos militares y diplomáticos. La historia de la criptografía es muy antigua y fascinante, para una revisión histórica no técnica de la criptografía puede consultarse [16].

2.1. Terminología

A continuación se enuncian algunos conceptos básicos que utilizamos a lo largo del trabajo.

Por *entidad* entendemos a alguien ó algo que envía, recibe o manipula información.

Un *intruso*, es aquella entidad que intenta hacerse pasar por el remitente autorizado o el destinatario autorizado.

Un canal es un medio que le permite a dos entidades comunicarse. Un canal inseguro es cuando un adversario puede leer, borrar, reordenar información.

La *criptografía* es el estudio de métodos matemáticos para enviar mensajes en forma disfrazada, de tal manera que sólo el destinatario autorizado pueda quitar el disfraz y leer el mensaje.

El mensaje original que queremos enviar, conocido como *texto claro* (plaintext), es convertido en un mensaje aparentemente sin sentido llamado *texto cifrado* (ciphertext). El texto claro y el texto cifrado suelen escribirse en algún alfabeto (generalmente, pero no siempre, son escritos en el mismo alfabeto), el cual consiste de un conjunto finito de N caracteres y lo denotamos por $\mathcal{A} = \{A_0, A_2, A_3, \ldots, A_{N-1}\}$.

Podemos dividir al texto claro y al texto cifrado en bloques iguales de longitud k, a los que llamamos *unidades de mensaje*, denotados por P y C respectivamente. Denotamos por $\mathcal{P}$ al conjunto de unidades de mensaje de texto claro y con $\mathcal{C}$ al conjunto de unidades de mensaje de texto cifrado.

Para un conjunto de claves $\mathcal{K}$, tenemos que la *transformación de cifrado* con *clave de cifrado* $e \in \mathcal{K}$ es una función biyectiva que convierte a $\mathcal{P}$ en $\mathcal{C}$, en símbolos:

$$\mathcal{E}_e : \mathcal{P} \longrightarrow \mathcal{C}.$$

Esto es, dada una unidad de mensaje cifrado C hay una y solamente una unidad de mensaje de texto claro P de la cual proviene.

Para recuperar el mensaje original aplicamos la *transformación de descifrado*

$$\mathcal{D}_d : \mathcal{C} \longrightarrow \mathcal{P},$$

donde d es la *clave de descifrado*, que debe cumplir $\mathcal{D}_d\mathcal{E}_e = \text{Identidad}$.

Al proceso de aplicar la transformación de cifrado $\mathcal{E}_e$ a cada elemento de $\mathcal{P}$ se le llama *cifrar* (enciphering) y al proceso inverso se le llama *descifrar* (deciphering).

Con estos elementos representamos a un *criptosistema* como:

$$\mathcal{P} \overset{E_e}{\rightarrow} \mathcal{C} \overset{D_d}{\rightarrow} \mathcal{P}.$$

A menudo, el término criptosistema se usa para referirse a una familia completa de tales transformaciones. A las claves e, d se les conoce como par de claves y se denota por (e, d). La seguridad de los criptosistemas depende en gran medida de la transformación de cifrado.

Una transformación de cifrado es frágil si una tercera entidad, sin conocimiento previo del par de claves (e, d), puede sistemáticamente recuperar el mensaje original.

Una transformación de cifrado puede romperse probando todas las posibles claves para ver cuales están utilizando las entidades comunicantes (asumimos que la transformación de cifrado es de conocimiento público); esto se conoce como la *búsqueda exhaustiva* en el conjunto de claves $\mathcal{K}$. Por consiguiente, se sigue que el número de claves en $\mathcal{K}$ debe ser suficientemente grande para hacer esta búsqueda difícil de realizar.

El *criptoanálisis* es el estudio de técnicas matemáticas para intentar frustar los métodos criptográficos. Aquellas personas que están dedicadas al criptoanálisis se le conoce como *criptoanalistas.*

Para romper un criptosistema, la estrategia usada por el criptoanalista depende de la naturaleza de éste y del conocimiento de ciertos parámetros que están ligados al criptosistema dado.

Al estudio de la criptografía y del criptoanálisis se le conoce como *criptología.*

Uno de los primeros pasos para implementar un criptosistema es *etiquetar* a los elementos de $\mathcal{P}$ y $\mathcal{C}$ con los elementos de un conjunto $\mathcal{B}$. Usaremos el símbolo $\approx$ para identificar a

los elementos de dos conjuntos dados, por ejemplo, si $\mathcal{A} = \{A_0, A_2, A_3, \ldots, A_{N-1}\}$ entonces $\mathcal{A} \approx \mathbb{Z}_N$; es decir, hacemos la identificación (etiquetamos) $A_0 = 0, A_1 = 1, \ldots, A_{N-1} = N - 1$.

En algunas ocasiones podemos utilizar otros elementos matemáticos para etiquetar las unidades de mensaje, por ejemplo, vectores ó puntos en alguna curva como lo veremos en el Capítulo 5.

2.2. Objetivos criptográficos

La criptografía está relacionada con los aspectos de la seguridad en la información. Cuando se habla de seguridad en la información se deben cumplir los siguientes objetivos:

1. *Privacidad,* mantener el contenido de la información fuera del alcance de las entidades no autorizadas, es decir, permitir a dos usuarios **A**(**A**le) y **B**(**B**ob) enviarse mensajes a través de un canal inseguro de tal forma que sólo puedan ser leídos por ellos; el contenido de la información se mantiene en secreto.

2. *Integridad,* no se permite la alteración del mensaje por entidades no autorizadas. Las alteraciones incluyen acciones tales como la inserción, borrado, repetición y reordenamiento de los caracteres.

3. *Autenticación,* se relaciona con la identificación de usuarios, es decir garantiza que las entidades sean quien dicen ser. No permite que la información sea manipulada por entidades no autorizadas.

4. *No repudio,* establecer un compromiso entre los participantes de una comunicación. Con esto se previene que **A** ó **B** nieguen haber escrito un mensaje.

2.3. Criptografía convencional

Históricamente, el principal objetivo de la criptografía era la privacidad, lo cual se logra al utilizar criptosistemas convencionales.

Un criptosistema convencional, también conocido como de *clave simétrica*, consiste de las funciones $\mathcal{E}_e$, $\mathcal{D}_d$ con $\mathcal{D}_d(\mathcal{E}_e(P)) = P$ para toda $P \in \mathcal{P}$, tal que para cada par de claves (e, d), es "computacionalmente fácil" determinar d conociendo e y determinar e de d.

Comentario 2.3.1 *El término "computacionalmente fácil" significa que podemos resolver un problema en tiempo polinomial y que puede atacarse usando recursos disponibles.*

Si el usuario **A** desea enviar un mensaje a **B**, debe enviar $C = \mathcal{E}_e(P)$ a **B**, al aplicar **B** la función $\mathcal{D}_d$ a C puede recuperar el mensaje original; pero antes deben ponerse de acuerdo sobre cuál va a ser la clave "secreta" (e, d).

En este tipo de criptosistemas el criptoanalista intentará descubrir el mensaje original sin el conocimiento del par de claves (e, d).

En la actualidad los criptosistemas convencionales son más utilizados para enviar mensajes debido a que es más fácil su implementación, sin embargo tienen las siguientes desventajas:

1. Problema de la distribución de claves: No está disponible un medio seguro para que las entidades puedan intercambiar la clave secreta.
2. Problema de la administración de claves: En una red de n usuarios, cada par de usuarios debe compartir una clave secreta, para un total de $C_2^n = \frac{n(n-1)}{2}$ claves secretas (C_2^n es el número de aristas de una gráfica no dirigida de n puntos). Si n es muy grande, entonces es difícil de administrar la cantidad total de claves secretas.
3. Problema de la autenticación: Es decir, que el usuario **B** esté convencido de que el mensaje fue escrito por **A**.

2.3.1. Transformaciones afines que definen criptosistemas

Vamos a empezar con el caso en el que el alfabeto $\mathcal{A}$ tiene N caracteres y las unidades de mensaje son bloques de un solo carácter.

Sea $\mathcal{P} = \mathcal{C} = \mathcal{A} \approx \mathbb{Z}_N$. Sea $a \in \mathbb{Z}_N^*$ y $b \in \mathbb{Z}_N$, por la Proposición 1.3.2 la transformación afín $f : \mathbb{Z}_N \longrightarrow \mathbb{Z}_N$ dada por $f(x) \equiv ax + b \bmod N$ es una función biyectiva. Además $f^{-1}(y) \equiv a^{-1}(y - b) \bmod N$, $y \in \mathbb{Z}_N$. Por lo tanto, f es una transformación de cifrado de $\mathcal{P}$ en $\mathcal{C}$ y f^{-1} es la transformación de descifrado.

En este tipo de transformación $\mathcal{P}$ y $\mathcal{C}$ permanecen fijos (porque N es fijo), pero la transformación de cifrado depende de los parámetros a, b. La clave de cifrado es $e = (a, b)$ y la de descifrado es $d = (a^{-1}, -a^{-1}b)$.

Veamos un ejemplo particular.

Ejemplo 2.3.2 *Sea $\mathcal{A} = \{A, \ldots, Z\}$ el alfabeto español y $\mathcal{P} = \mathcal{C} \approx \mathbb{Z}_N$ donde $N = 27$. Si queremos cifrar el mensage*

$$\text{“}ESTEMENSAJENOESSEGURO\text{”}$$

usando la transformación anterior con $a = 7$, $b = 12$, obtenemos:

$$\begin{aligned} & 4\ 19\ 20\ 4\ 12\ 4\ 13\ 19\ 0\ 9\ 4\ 13\ 15\ 4\ 19\ 19\ 4\ 6\ 21\ 18\ 15 \\ \longmapsto\ & 13\ 10\ 17\ 13\ 15\ 13\ 22\ 10\ 12\ 21\ 13\ 22\ 9\ 13\ 10\ 10\ 13\ 0\ 24\ 3\ 9 \\ & = \text{“}NKQNONVKMUNVJNKKNAXDJ\text{”}. \end{aligned}$$

Alguien que quiera descifrar este mensaje, utilizará la función inversa con la clave de descifrado $(4, 6)$.

Debemos notar que esto funciona si y sólo si $(a, N) = 1$; ya que si $(a, N) > 1$, entonces podemos ver que más de una unidad de texto claro dará como resultado la misma unidad de texto cifrado. Por definición, esta no es una transformación de cifrado.

Ahora supongamos que hemos interceptado un mensaje grande de texto cifrado utilizando bloques de un sólo carácter utilizando la transformación afín anterior. Mediante el análisis de frecuencias podemos determinar la clave de cifrado $e = (a, b)$ y lo explicamos mediante el siguiente ejemplo.

Ejemplo 2.3.3 *Supongamos que en nuestro texto cifrado el carácter que más aparece es "N", mientras que el segundo carácter que más aparece es "M". Entonces es razonable pensar que estos caracteres son los cifrados de "E" y "A" respectivamente, que son las letras que más aparecen en el alfabeto español. De esta manera, al reemplazar a los caracteres por sus equivalentes numéricos y al sustituirlos en* $P \equiv a'C + b' \ mod\ N$, *obtenemos*

$$13a' + b' \equiv 4 \ mod\ 27$$
$$12a' + b' \equiv 0 \ mod\ 27$$

Una forma rápida para resolver este sistema es restar las dos congruencias para eliminar b'. *Al resolverlo tenemos que* $a' \equiv 4 \ mod\ 27$ *y* $b' \equiv 4 - 13a' \equiv 6 \ mod\ 27$. *Así que el mensaje puede descifrarse con la fórmula* $P \equiv 4C + 6 \ mod\ 27$.

Ahora supongamos que nuestras unidades de texto claro y de texto cifrado son bloques de k caracteres en $\mathcal{A}$, es decir $P \in \mathcal{P} = \mathcal{C}$ si $P = B_k B_{k-1} \ldots B_1$ con $B_i \in \mathcal{A}, i = 1, \ldots, k$.

Sea $B_i \in \mathbb{Z}_N \approx \mathcal{A}$, consideramos al bloque $B_k B_{k-1} \ldots B_1$ como un número de k dígitos en base N, es decir

$$B_k B_{k-1} \ldots B_1 = B_1 N^0 + B_2 N^1 + \ldots + B_k N^{k-1} \in \{0, \ldots, N^k - 1\} = \mathbb{Z}_{N^k}.$$

Por lo tanto $\mathcal{P} = \mathcal{C} \approx \mathbb{Z}_{N^k}$.

Sea $a \in \mathbb{Z}_{N^k}^*$ tal que $(a, N^k) = 1$ (que es equivalente a $(a, N) = 1$) y $b \in \mathbb{Z}_{N^k}$. La transformación afín $f : \mathbb{Z}_{N^k} \longrightarrow \mathbb{Z}_{N^k}$ dada por $f(x) \equiv ax + b \mod N^k$ es una transformación de cifrado de $\mathcal{P} \approx \mathbb{Z}_{N^k}$ en $\mathcal{C} \approx \mathbb{Z}_{N^k}$. La transformación de descifrado es $f^{-1}(y) \equiv a^{-1}(y - b) \mod N^k$.

A continuación tenemos un ejemplo práctico en donde nuestras unidades de cifrado son bloques de dos letras llamado *digrafos.*

Ejemplo 2.3.4 *Sea* $\mathcal{A} = \{A, B, C, \ldots, Z, \square\} \approx \mathbb{Z}_{28}$, $k = 2$, *entonces* $\mathcal{P} = \mathcal{C} \approx \mathbb{Z}_{28^2}$. *Para cifrar el mensaje*

$$\text{"}COMERCIOELECTRONICO\text{"},$$

dividimos el mensaje en bloques de dos y completamos el último bloque con □. *Sea* $a = 25$, $b = 9$, *obtenemos:*

$$\begin{aligned} &71\ 340\ 506\ 239\ 123\ 114\ 578\ 433\ 226\ 447 \\ \longmapsto\ &216\ 669\ 115\ 496\ 732\ 507\ 347\ 642\ 171\ 208 \\ = \text{“}&HT\ WY\ ED\ QT\ ZE\ \tilde{N}O\ ML\ VZ\ GD\ HM\text{”}. \end{aligned}$$

Calculando 25^{-1} *mod* 784 *obtenemos la clave de descifrado, para este ejemplo* $d = (345, 753)$.

Para romper un sistema de cifrado digráfico usando la transformación afín $f(x) \equiv ax + b \bmod N^2$ necesitamos conocer el texto cifrado correspondiente a dos unidades de mensaje de texto claro diferente. Ya que las unidades de mensaje son digrafos, un análisis de frecuencia equivale a contar los bloques de dos caracteres que más frecuentemente aparecen en una cadena grande de texto cifrado (ignoramos aquéllos caracteres que nos sirven para completar los digráfos) y compararlo con un análisis de frecuencia conocido. Por ejemplo, si usamos el alfabeto inglés, un análisis estadístico muestra que "TH" y "HE" son los digráfos que más aparecen (en ese orden) entonces frecuentemente (pero no siempre) se puede determinar la clave de cifrado.

Es un poco más difícil romper el criptosistema utilizando el análisis de frecuencia para $k = 2$, por lo que sería más seguro utilizar bloques con $k > 3$. Sin embargo, si aumentamos el número de bloques debemos tener en cuenta el tiempo que nos tomaría en realizar las operaciones aritméticas para realizar el cifrado y descifrado de mensajes, la operación más importante con la que nos encontraríamos sería la de calcular a^{-1} por medio del algoritmo extendido de Euclídes.

2.3.2. Matrices de cifrado.

Sea $\mathcal{A}$ un alfabeto de N caracteres, $\mathcal{P} = \mathcal{C} = \mathcal{A} \times \mathcal{A} \approx \mathbb{Z}_N \times \mathbb{Z}_N$ y $A \in M_{2\times 2}(\mathbb{Z}_N)$ tal que $(\det A, N) = 1$. Por la Proposición 1.3.3 la transformación $f : \mathbb{Z}_N \times \mathbb{Z}_N \longrightarrow \mathbb{Z}_N \times \mathbb{Z}_N$ dada por $TP = AP$ es biyectiva, por lo tanto tenemos un criptosistema

$$\mathcal{P} \xrightarrow{T} \mathcal{C} \xrightarrow{T^{-1}} \mathcal{P}$$

Para cifrar una secuencia de k digrafos $P = P_1 P_2 \ldots P_k$ escribimos los k vectores como una matriz de $2 \times k$, la cual también denotamos por P, y calculamos AP.

En este caso la matriz A es la clave de cifrado y para alguien que quiera conocer el mensaje original tendrá que usar la clave de descifrado A^{-1}.

Ejemplo 2.3.5 *Sean* $\mathcal{A} = \{A, \ldots, Z, \square\} \approx \mathbb{Z}_{28}$, $\mathcal{P} = \mathcal{C} = \mathcal{A} \times \mathcal{A} \approx \mathbb{Z}_{28} \times \mathbb{Z}_{28}$, *y* $A = \begin{pmatrix} 2 & 3 \\ 7 & 9 \end{pmatrix}$. *Al mensaje “COMERCIOELECTRONICO” le asociamos la matriz* $P = \begin{pmatrix} 2 & 12 & 18 & 8 & 4 & 4 & 20 & 15 & 8 & 15 \\ 15 & 4 & 2 & 15 & 11 & 2 & 18 & 13 & 2 & 27 \end{pmatrix} \in M_{2\times 10}(\mathbb{Z}_{28})$. *Por lo tanto,*

$$AP = \begin{pmatrix} 21 & 8 & 14 & 5 & 13 & 14 & 10 & 13 & 22 & 27 \\ 9 & 8 & 4 & 23 & 15 & 18 & 22 & 26 & 18 & 12 \end{pmatrix}$$
$$= \text{“}UJIIÑEFWNOÑRKVNZVR\square M\text{”}.$$

Para este ejemplo $A^{-1} = \begin{pmatrix} 25 & 1 \\ 21 & 18 \end{pmatrix}$.

El sistema convencional más utilizado es el DES (Data Encryption Standar). Fue desarrollado por IBM y posteriormente adoptado como un estándar en los E.U en 1977. El DES cifra la información en bloques de 64 bits, utiliza una clave K de 56 bits para obtener un texto cifrado de 64 bits. El algoritmo se puede consultar en [43, pag 70-81]. El DES es muy fácil de implementar tanto en hardware como en software.

2.4. Teoría de la complejidad

Antes de continuar con el Capítulo 3 discutimos una notación conveniente para resumir la situación con los tiempos estimados.

El objetivo principal de la teoría de la complejidad es proporcionar mecanismos para clasificar los problemas según los recursos que se necesitan para resolverlos. La clasificación no debe depender de un modelo computacional particular, sino que debe medir la dificultad intrínseca del problema. Los recursos medidos pueden incluir tiempo, espacio de almacenamiento, número de procesadores, etc., pero generalmente el objetivo es medir el tiempo y algunas veces el espacio.

Definición 2.4.1 *Sean* $f, g : \mathbb{N} \to \mathbb{R}^+$ *funciones. Decimos que* f *tiene un comportamiento asintótico del orden de* g *ó que* f *es “O grande” de* g *y se denota* $f(n) = O(g(n))$, *si existe una constante* $c > 0$ *y* $n_0 \in \mathbb{N}$ *tal que* $f(n) \leq cg(n)$ *si* $n \geq n_0$.
Por otro lado, decimos que f *es “o pequeña” de* g *y se denota* $f(n) = o(g(n))$, *si* $lím_{n\to\infty} f(n)/g(n) = 0$.

De esta manera, $o(1)$ se utiliza para representar a una función cuyo límite es cero cuando n tiende a infinito.

Definición 2.4.2 *Un algoritmo es un procedimiento computacional “bien definido” que toma algún valor como entrada, se detiene y produce otro valor como salida.*

Un algoritmo es de *tiempo polinomial* cuando su complejidad es $O(n^c)$ para alguna constante $c \in \mathbb{R}^+$, donde n es la longitud de bits de entrada y c es independiente de n. En general, estos son los algoritmos deseables debido a que son más rápidos.

Por otro lado, los algoritmos con complejidad $O(c^{f(n)})$, donde $c > 1$ es una constante y f es un polinomio en $n \in \mathbb{N}$, son algoritmos de *tiempo exponencial.*

Un algoritmo de tiempo subexponencial es aquél que tiene la siguiente complejidad:

$$L_n(r,c) = O\left(\exp\left((c+o(1))\,(ln\ n)^r\,(lnln\ n)^{1-r}\right)\right).$$

donde $r \in \mathbb{R}$ con $0 \leq r < 1$ y c es una constante positiva. Los algoritmos subexponenciales son más rápidos que los algoritmos de tiempo exponencial pero más lentos que los algoritmos polinomiales. Burdamente hablando, los algoritmos exponenciales son algoritmos *ineficientes.*

Capítulo 3

Criptografía de clave pública

En 1976 W. Diffie y M. Hellman [8] propusieron un tipo de criptosistema completamente diferente a los descritos en el Capítulo 2, dando lugar a la *criptografía de clave pública*. Desde entonces, se han hecho varios intentos para encontrar criptosistemas prácticos de clave pública basados en la aparente intratabilidad computacional para resolver algunos problemas matemáticos como: la factorización de enteros grandes, el cálculo de logaritmos discretos sobre campos finitos, el problema de la mochila y el cálculo de logaritmos discretos sobre curvas elípticas.

3.1. La idea de la criptografía de clave pública

Para introducir el concepto de criptografía de clave pública, primero definimos el concepto de función de un sentido (one-way function) y el de función tramposa (trapdoor function).

Definición 3.1.1 *Una función de un sentido $f : \mathcal{P} \to \mathcal{C}$ es una función invertible tal que para cada $P \in \mathcal{P}$ es fácil calcular $f(P)$, mientras que para la mayoría de $C \in \mathcal{C}$ es "difícil" calcular $f^{-1}(C)$.*

Entendemos por "difícil" el requerir que sea calculable en tiempo exponencial. En términos prácticos, "difícil" significa *computacionalmente no factible*, es decir, no es posible usar los mejores algoritmos conocidos y la mejor tecnología computacional disponible para obtener $f^{-1}(C)$.

Veamos como funciona el sistema de passwords. Los passwords de los usuarios son valores x en el dominio de una función f. Para mantener la lista de passwords fuera del alcance de los intrusos, la computadora no almacena estos passwords x. Más bien, almacena el valor $f(x)$ para el usuario con password x. Cuando el usuario **A** quiere entrar al sistema introduce x, la computadora calcula $f(x)$ y lo compara con el valor almacenado $y = f(x)$, le otorga el acceso y luego borra cualquier registro de x. Además, el valor $f(x)$

está expuesto al proceso de cifrado. En caso de que un malhechor obtenga una copia de los passwords cifrados, no podrá utilizarlos inmediatamente, ya que primero tendrá que descifrarlos y aún cuando los detalles de la función de cifrado estuvieran disponibles, el proceso de descifrado le tomaría mucho tiempo.

Una función de un sentido $f : \mathcal{P} \rightarrow \mathcal{C}$ es una *función tramposa* (trapdoor function) si existe información adicional con la cual f puede invertirse eficientemente. A esta información adicional se le llama "trampa" (trapdoor).

Así, un criptosistema de clave pública es una familia de funciones tramposas $\{f\}$, en donde $f : \mathcal{P} \rightarrow \mathcal{C}$ es fácil de calcular una vez que se conoce la clave de cifrado K_E y difícil de calcular la función inversa $f^{-1} : \mathcal{C} \rightarrow \mathcal{P}$ sin alguna información adicional (la clave de descifrado K_D). No es computacionalmente factible determinar K_D a partir de K_E.

Por definición, un criptosistema de clave pública tiene la propiedad de que es difícil determinar K_D a partir del conocimiento de K_E; es decir, alguien que sólo sabe cifrar no puede usar la clave de cifrado para calcular la clave de descifrado sin tener que realizar una gran cantidad de cálculos. El nombre de *clave pública* se debe a ésta propiedad, la cual hace posible que la clave de cifrado de un usuario **A,** que denotamos por $K_{E,A}$, sea de conocimiento público sin comprometer la seguridad de su clave de descifrado $K_{D,A}$. De esta manera alguien que quiera enviarle mensajes cifrados al usuario **A**, sólo requerirá de una copia auténtica de la clave pública de **A**.

En este tipo de criptosistemas es común denominar a la clave de cifrado K_E como *clave pública* y a la clave de descifrado K_D como *clave privada.* La razón de lo anterior, es que de esta forma se reserva el término *clave secreta* para los criposistemas convencionales.

Las definiciones de criptosistema de clave pública, función tramposa ó función de un sentido no son precisas desde el punto de vista matemático, ya que depende de los avances en la tecnología computacional y el descubrimiento de nuevos algoritmos que aceleren la ejecución de las operaciones aritméticas. Debido a esto, es posible que una transformación de cifrado que es considerada una función de un sentido o tramposa en el año 1995, puede que pierda su estatus en el año 2007 o en años posteriores.

3.2. Almacenamiento y distribución de claves

En la práctica, los criptosistemas de clave pública para enviar mensajes tienden a ser más lentos que los sistemas convencionales; el número de unidades de mensaje por segundo que pueden ser enviados es menor. Sin embargo, si los usuarios de una red quieren seguir utilizando criptosistemas convencionales, pueden utilizar criptosistemas de clave pública para intercambiar sus claves secretas $K = (e, d)$. Ahora con este recurso las claves pueden cambiarse periódicamente, mientras que los grandes volúmenes de mensajes pueden cifrarse utilizando métodos convencionales.

Cuando se trabaja con varias claves de longitud pequeña es posible memorizarlas sin necesidad de recurrir a ningún sistema para su almacenamiento, pero cuando se deben utilizar diversas claves para cifrar información se hace imprescindible su almacenamiento.

En los criptosistemas de clave pública el problema está resuelto, puesto que el usuario sólo debe mantener en secreto su clave privada, estando la clave pública en una base de datos ó en directorios de claves públicas que alguna entidad confiable se encarga de guardar. Dicha entidad es una autoridad certificadora (AC) con su propio par de claves $(K_{E,AC}, K_{D,AC})$. Veamos a grandes rasgos como funciona esto.

Supongamos que estamos seguros de que todos los usuarios obtienen una copia auténtica de $K_{E,AC}$. Entonces cada usuario **A** del sistema debe contactar a la AC, identificarse de alguna forma con la AC y entonces enviarle su clave pública $K_{E,A}$. Si la AC acepta la identidad del usuario, generará el certificado que consiste de una cadena ID que identifique al usuario, la clave pública y la firma de la AC (Sección 3.6).

3.3. El problema del logaritmo discreto

Cuando trabajamos con números reales, calcular $y = b^x$ (b fijo) con una precisión predeterminada no es significativamente más fácil que calcular $x = \log_b y$ con una precisión predeterminada.

Ahora supongamos que tenemos un grupo finito. Supongamos que b es un elemento fijo del grupo finito, al cual denominamos base. Usando el método de cuadrados repetidos (sección 1.1.3) podemos calcular b^x, para x grande, de una manera bastante rápida en tiempo (es de crecimiento polinomial en $log\ x$).

Sin embargo supongamos que sabemos que un elemento dado y en el grupo finito es de la forma $y = b^x$, ¿cómo podemos encontrar la potencia de b que da como resultado y?; es decir, ¿cómo calculamos $x = \log_b y$?. A este cuestionamiento se le llama el problema del logaritmo discreto. A x se le llama el logaritmo discreto de y a la base b. La palabra "discreto" distingue el caso del grupo finito del caso clásico (continuo).

Definición 3.3.1 *Sea $\mathcal{G}$ un grupo finito cíclico de orden n. Sean $b, y \in \mathcal{G}$. El logaritmo discreto de y a la base b, denotado por $log_b y$, es el único entero x, $0 \leq x \leq n-1$ tal que $y = b^x$.*

El hecho de *encontrar* el entero x se le conoce como el *problema generalizado del logaritmo discreto* (**PGLD**). Un caso particular es el siguiente:

Definición 3.3.2 *El problema del logaritmo discreto (**PLD**) en $\mathbb{F}_q^*$ es el siguiente: dado $y \in \mathbb{F}_q^*$ y b un generador de $\mathbb{F}_q$, encontrar el entero x, $1 \leq x \leq q-1$ tal que $b^x = y$.*

Ejemplo 3.3.3 *Sea* $\mathcal{G} = \mathbb{F}_{19}^*$, *entonces* $b = 2$ *es un generador de* $\mathcal{G}$. *En este caso el logaritmo discreto de 7 base 2 es 6, es decir* $\log_2 7 = 6$.

Aún más, se puede *generalizar* el **PGLD** al requerir que el grupo $\mathcal{G}$ no sea cíclico, si es así, no se requiere que b sea un generador de $\mathcal{G}$. Se cree que este problema es mucho más difícil de resolver [28, pag. 104].

Los grupos de mayor interés en criptografía son: el grupo multiplicativo $\mathbb{F}_q^*$ del campo finito $\mathbb{F}_q$ con $q = p^n$, p primo; incluyendo los casos particulares $\mathbb{F}_p^*$ del campo finito $\mathbb{F}_p$ y $\mathbb{F}_{2^n}^*$ del campo finito $\mathbb{F}_{2^n}$. También es de interés el grupo que forman los puntos de una curva elíptica sobre campos finitos.

3.3.1. Ataques conocidos al problema del logaritmo discreto

Algunos de los algoritmos conocidos para resolver el problema del logaritmo discreto son: Pollard's rho, Pohlig-Hellman y el Cálculo de Indices con sus variantes: el algoritmo de Coppersmitth y el Number Field Sieve (adaptado para el cálculo de logaritmos discretos), para una revisión de estos algoritmos puede consultarse [28, pag. 106-113].

En [15] Joux y Lercier calcularon logaritmos discretos en el campo $\mathbb{F}_p$ con $p = 120$ dígitos decimales, es decir, 399 bits. Por esta razón, en los criptosistemas basados en el problema del logaritmo discreto sobre campos $\mathbb{F}_p$, p no debe ser menor a 399 bits. Se recomienda que para tener buena seguridad en éstos criptosistemas p debe ser de al menos 1024 bits.

Para el caso de los campos de característica 2, el trabajo más reciente se debe a Emmanuel Thomé [44] para calcular logaritmos discretos en el campo $\mathbb{F}_{2^{607}}$ ($n = 607$ bits) al usar el algoritmo para Cálculo de Indices de Coppersmith [7], el cuál tiene complejidad subexponencial con respecto al orden del campo.

3.4. Sistema de distribución de claves Diffie-Hellman

En el artículo [8], Diffie y Hellman propusieron un protocolo en donde dos usuarios, **A** y **B**, pueden originar y compartir información secreta a través de un medio de comunicación inseguro. Este protocolo está basado en el problema del logaritmo discreto.

Ahora describimos el método Diffie-Hellman para la distribución de claves. Supongamos que q es de conocimiento público y que $g \in \mathbb{F}_q$ es un elemento fijo público. Idealmente g debe ser un generador de $\mathbb{F}_q$ aunque no es necesario; así, si queremos que nuestro elemento aleatorio de $\mathbb{F}_q^*$ tenga la oportunidad de ser cualquier elemento, g debe ser un generador.

Comentario 3.4.1 *Cuando nos referimos al término "aleatorio" significa que el número fue elegido con la ayuda de un generador de números aleatorios (ver [37]); es decir, un programa de computadora que genera una secuencia de dígitos de tal forma que no se puedan predecir o duplicar.*

Supongamos que los usuarios **A** y **B** quieren comunicarse por medio de un criptosistema convencional, pero antes deben ponerse de acuerdo sobre una clave en común (un elemento aleatorio de $\mathbb{F}_q^*$). Entonces **A**le elige un entero aleatorio a entre 1 y $q-1$, que mantendrá en secreto, calcula $y_A = g^a \in \mathbb{F}_q$ para luego hacerlo público.

Notamos que alguien que quiera conocer la clave secreta a debe calcular el logaritmo discreto de y_A:

$$a = \log_g y_A \text{ para } 1 \leq y_A \leq q-1$$

De la misma forma, **B**ob elige y mantiene en secreto un elemento aleatorio b y publica $y_B = g^b \in \mathbb{F}_q$. Entonces la clave secreta en común que utilizarán es:

$$K_{AB} = g^{ab}.$$

Ambos usuarios pueden calcular esta clave. Por ejemplo, **A**le conoce g^b (que es de conocimiento público) y su propia clave secreta a, de donde calcula $(g^b)^a = g^{ba} = g^{ab}$. Cualquier otro usuario tendría que calcular

$$K_{AB} = g^{a(\log_g y_B)}.$$

Si la siguiente suposición se cumple para el grupo multiplicativo $\mathbb{F}_q^*$ entonces una tercera persona no autorizada, quien conoce únicamente g^a y g^b, será incapaz de determinar la clave K_{AB}.

3.4.1. La suposición de Diffie-Hellman

Dado un generador g de $\mathbb{F}_q^*$ y dos elementos $g^a, g^b \in \mathbb{F}_q^*$, no es computacionalmente factible calcular g^{ab} conociendo simplemente g^a y g^b. En otras palabras, nadie puede imaginar una manera de pasar de g^a y g^b a g^{ab} sin determinar primero a ó b; pero tal forma podría existir.

La suposición de Diffie-Hellman (ver [8]) es a priori tan fuerte como la suposición de que no es computacionalmente factible calcular el logaritmo discreto de elementos del grupo. Esto es, si fuera posible calcular fácilmente los logaritmos discretos entonces obviamente falla la suposición de Diffie-Hellman. A continuación describimos un ejemplo práctico.

Ejemplo 3.4.2 *Sea* $q = 517$ *y* $g = 23$. ***A*** *y* ***B*** *eligen los números* $a = 8$ *y* $b = 14$ *respectivamente.* ***A*** *calcula* $23^8 \ mod\ 517 = 56$, ***B*** *calcula* $23^{14} \ mod\ 517 = 89$. *Y publican los resultados 56 y 89. Entonces* ***A****le calcula* $89^8 \ mod\ 517 = 243$, *mientras que* ***B****ob calcula* $56^{14} \ mod\ 517 = 243$. *Tenemos entonces que la clave que comparten es* 243 *y que podrán utilizarla en futuras comunicaciones.*

Observamos que esto no es un intercambio de clave autentificada ya que una tercera entidad **C** podría hacerse pasar por **A** ó **B**. Sin embargo el protocolo puede modificarse al requerir una entidad confiable para certificar (firmar).

3.5. Funciones hash

Una función hash reduce cualquier cadena de longitud arbitraria a otra de longitud fija (por ejemplo, de una cadena de 10^6 a una cadena de 150 o 200 bits) y se usa para resolver el problema de la integridad de los mensajes, así como para firmar digitalmente documentos que generalmente son demasiados grandes.

En lo sucesivo, la función hash se denotará por H y la información que va a ser protegida con x. La imagen de x bajo la función hash H es $H(x)$. Se asume que la función hash es de conocimiento público (cualquier persona puede calcular $H(x)$ para cualquier mensaje x). Una segunda suposición es que debe ser fácil el cálculo del valor hash $h = H(x)$.

El concepto de función hash de un sentido (OWHF, por sus siglas en inglés) fue presentada por Diffie y Hellman. Una función hash de un sentido es una función H que satisface las siguientes condiciones:

1. El argumento x puede ser de longitud *arbitraria* y el resultado h tiene una longitud *fija* de n bits (con $n \geq 64, \ldots, 80$ bits).

2. La función hash debe ser de *un sentido,* en el entendido de que dada una y en el contradominio de H, es difícil encontrar un mensaje x tal que $H(x) = y$ (preimagen resistente) y dada una x en el dominio de H es difícil encontrar un mensaje $x' \neq x$ tal que $H(x') = H(x)$ (segunda preimagen resistente).

La definición formal de una función hash de una vía la podemos encontrar en ([35], pag. 160).

Los objetivos de un adversario para atacar una función hash de un sentido serían: dada una y, encontrar x tal que $H(x) = y$ ó dado el par $(x, H(x))$ encontrar $x' \neq x$ tal que $H(x') = H(x)$.

Ejemplos de funciones hash son el SHA-1 [11], RIPEMD 160 [9] y MD5 [36].

3.6. Firma digital

Una firma digital es equivalente a una firma convencional, pero con la ventaja de que la firma digital no se puede falsificar. Cuando un usuario **A** quiera enviarle un mensaje

m al usuario **B** debe añadirle una "huella", la cual puede ser el valor hash h del mensaje m (sección 3.5), fecha y hora en la que se envió el mensaje, identificación del emisor, etc.

De manera simple, una firma digital se define como el cifrado de la huella utilizando la clave privada de la persona que envía el mensaje. La definición formal de una firma digital (también conocido como esquema de firma digital) se puede consultar en [43, pag. 203].

En criptografía de clave pública existe una manera fácil de resolver el problema de la autenticación, la cual se logra a través de la firma digital. El uso de las firmas digitales también proporcionan los servicios de integridad y no repudio.

Supongamos que **A**le desea enviarle un mensaje firmado y cifrado a **B**ob. Por simplicidad, supongamos que $\mathcal{P} = \mathcal{C}$ y que son las mismas para todos los usuarios. Sea f_A la transformación de cifrado con la que cualquier usuario del sistema puede enviarle mensajes a **A** y f_B lo análogo para **B**. Sea h_A la "huella" de **A**. Podría no ser suficiente para **A** enviarle el mensaje cifrado $f_B(h_A)$ a **B**, ya que todo mundo sabe como hacerlo, así que no habría forma de saber si la huella fue alterada. Entonces **A** opta por trasmitirle $f_B f_A^{-1}(h_A)$ al principio ó al final del mensaje cifrado, cuando **B** descifra todo el mensaje al aplicar f_B^{-1}, incluyendo ésta parte, él encuentra que todo se ha convertido en texto claro excepto por una pequeña parte de caracteres raros, que es $f_A^{-1}(h_A)$. Dado que **B** cree que el mensaje se lo envió **A**, él aplica f_A (que es de conocimiento público) y obtiene h_A. Entonces **B** comprueba que el mensaje lo originó **A**, ya que sólo **A** conoce f_A^{-1} (clave privada). De esta forma se puede evitar que alguien suplante a un usuario y envíe mensajes falsos a otros usuarios, solucionando de esta manera el problema de la autenticación.

En particular, si la huella h_A de **A** consiste del código hash $h = H(x)$, donde x es el texto completo de su mensaje, entonces **B** no sólo puede verificar que el mensaje fué enviado por **A** sino que no fué adulterado durante la transmisión. Es decir, **B** aplica la función hash H al texto claro descifrado y checa que el resultado concuerda con el valor hash h_A de **A**. El hecho de aplicarle una función hash H a un mensaje ahorra tiempo y espacio.

Un método tradicional para autenticar a los usuario es por medio de contraseñas, los cuales se utilizan para solicitar el acceso a un sistema en particular. Un ejemplo práctico lo podemos encontrar al momento de generar contraseñas en el sistema operativo UNIX, que consta de 13 caracteres que incluyen letras minúsculas y mayúsculas, números, el punto y la barra de directorios (/). Los dos primeros caracteres se denominan *salt* y se generan a partir del contador del reloj en el momento que se asigna la contraseña, con lo cual una misma contraseña puede aparecer cifrada 64^2 formas diferentes. Cuando se quiera accesar al sistema se compara la contraseña introducida con la almacenada en el fichero. Sin embargo las contraseñas no proporcionan los servicios de integridad, confidencialidad y no-repudio.

3.7. El criptosistema RSA

En 1978 **R**ivest, **S**hamir y **A**dleman [1] propusieron una implementación de una función tramposa; el éxito de este criptosistema se basa en la dificultad de la factorización de un número que es el producto de dos números primos grandes.

Sean p, q números primos grandes elegidos de manera aleatoria, $n = pq$; sea e elegido de manera aleatoria con $1 < e < \varphi(n) = (p-1)(q-1)$ y $(e, \varphi(n)) = 1$.

La función tramposa propuesta es $f(P) \equiv P^e \bmod n$. La información tramposa correspondiente es el entero d que satisface $e \cdot d \equiv 1 \bmod \varphi(n)$, es decir, $d \equiv e^{-1} \bmod \varphi(n)$ (d es el inverso multiplicativo de e mod $\varphi(n)$).

En el Capítulo 1 vimos que si n es el producto de dos primos p y q, el conocimiento de éstos primos es equivalente al conocimiento de $\varphi(n)$, lo cual implica que p y q deben mantenerse en secreto.

Comentario 3.7.1 *Al elegir p y q, el usuario **A** debe tener en cuenta ciertas condiciones. Las más importantes son: que los dos números primos no estén demasiado cerca (por ejemplo, p debe ser unos cuantos dígitos decimales más grande que q); que $p-1$ y $q-1$ tengan un máximo común divisor bastante pequeño y que ambos tengan al menos un factor primo grande.*

De esta manera, el usuario **A** elige dos números primos p_A, q_A y un número aleatorio e_A el cual no tiene factor común con $(p_A - 1)(q_A - 1)$. Luego **A** calcula $n_A = p_A q_A$, $\varphi(n_A) = n_A + 1 - p_A - q_A$, y también $d_A = e_A^{-1} \bmod \varphi(n_A)$.

Entonces **A** publica la clave de cifrado $K_{E,A} = (n_A, e_A)$ y mantiene oculta la clave de descifrado $K_{D,A} = (n_A, d_A)$. La transformación de cifrado $f : \mathbb{Z}_{n_A} \longrightarrow \mathbb{Z}_{n_A}$ está dada por $f(P) \equiv P^{e_A} \bmod n_A$ y la transformación de descifrado $f^{-1} : \mathbb{Z}_{n_A} \longrightarrow \mathbb{Z}_{n_A}$ está dada por $f^{-1}(C) \equiv C^{d_A} \bmod n_A$.

Proposición 3.7.2 *Sea (n, e) la clave pública. Dada la clave privada d, uno puede obtener la factorización $n = pq$. Reciprocamente, dada la factorización de n se puede calcular d de manera eficiente.*

Las operaciones de cifrado son operaciones inversas puesto que $e \cdot d \equiv 1 \bmod \varphi(n)$ implica que $e \cdot d = t\varphi(n) + 1$ para algún entero $t \geq 1$. Si $P \in \mathbb{Z}_n$ entonces

$$\begin{aligned} (P^e)^d &\equiv P^{t\varphi(n)+1} \bmod n, \\ &\equiv (P^{\varphi(n)})^t P \bmod n, \\ &\equiv 1^t P \bmod n, \text{ por la Proposición 1.1.21,} \\ &\equiv P \bmod n. \end{aligned} \tag{3.1}$$

En la práctica, nos gustaría trabajar con $\mathcal{P}$ y $\mathcal{C}$ de manera uniforme a través del sistema. Por ejemplo, supongamos que estamos trabajando con un alfabeto de N caracteres. Sean k, l enteros positivos con $k < l$ tal que, por ejemplo, N^k y N^l tenga aproximadamente 200 dígitos decimales. Tomamos como nuestras unidades de mensaje de texto claro todos los bloques de k caracteres, los cuales consideramos como enteros de k dígitos en base N, es decir, les asignamos equivalentes numéricos entre 0 y $N^k - 1$. De la misma manera, consideramos nuestras unidades de texto cifrado como bloques de l letras en el mismo alfabeto. Luego, cada usuario debe elegir p_A y q_A tal que $n_A = p_A q_A$ cumpla que $N^k < n_A < N^l$. Entonces a cualquier unidad de texto claro –un entero menor que N^k– le corresponde un elemento en $\mathbb{Z}_{n_A}$ (para cualquier n_A del usuario), y dado que $n_A < N^l$ la imagen $f(P) \in \mathbb{Z}_{n_A}$ puede escribirse de manera única como un bloque de l caracteres. A continuación describimos como cifrar nuestras unidades de texto claro.

Para enviarle un mensaje al usuario **A**, **B**ob obtiene la clave pública $K_{E,A} = (n_A, e_A)$ de **A**le y luego hace la identificación $\mathcal{P} \approx \mathbb{Z}_{N^k}$. Posteriormente cifra cada $P \in \mathcal{P}$ por separado al calcular

$$C \equiv P^{e_A} \bmod n_A$$

usando el método de cuadrados repetidos visto en la sección 1.1.3, y envía cada $C \in \mathcal{C}$ a **A**le.

Una vez que ella recibe C, **A**le realiza lo siguiente

$$P \equiv C^{d_A} \bmod n_A.$$

Consideremos el siguiente ejemplo práctico.

Ejemplo 3.7.3 *Supongamos que **A**le elige los número primos $p_A = 281$ y $q_A = 167$ entonces $n_A = 46\,927$ y $\varphi(n_A) = 280 \cdot 166 = 46\,480$. Luego **A** escoge un número aleatorio $e_A = 39423$ tal que $(e_A, \varphi(n_A)) = 1$. Usando el algoritmo extendido de Euclídes (Sección 1.1.1) **A**le encuentra que $d_A = e_A^{-1}$ mod $46\,480 = 26767$. Los números p_A, q_A, d_A permanecen secretos. Ahora elegimos $N = 27$, $k = 3$, $l = 4$. Esto es, el texto claro consiste de bloques de 3 caracteres, mientras que el texto cifrado consiste de bloques de 4 caracteres en nuestro alfabeto de 27 caracteres. Para enviarle el mensaje "ECO" al usuario **A** con la clave de cifrado $(n_A, e_A) = (46927, 39423)$, primero encontramos los equivalentes numéricos de "ECO", a saber: $4 \cdot 27^2 + 2 \cdot 27 + 15 \cdot 27^0 = 2985$ y calculamos 2985^{39423} mod 46927, que es $7618 = 0 \cdot 27^3 + 10 \cdot 27^2 + 12 \cdot 27^1 + 4 \cdot 27^0 =$ "AKME". El usuario **A** sabe que la clave de descifrado $(n_A, d_A) = (46927, 26767)$ y con esto puede calcular 7618^{26767} mod $46\,927 = 2985 =$ "ECO".*

Un ataque evidente al criptosistema RSA es que un criptoanalista puede intentar factorizar n. Si logra hacer esto, entonces es fácil de calcular $\varphi(n) = (p-1)(q-1) = n+1-p-q$ y entonces puede calcular d a partir de e, exactamente como lo hizo **A**le. Se conjetura que romper el criptosistema RSA es computacionalmente equivalente a factorizar n (que aún es un problema abierto).

En la actualidad es recomendable utilizar claves de 1024 bits, es decir n debe ser de 308 dígitos. Esto es porque el número de bits en la representación binaria es $\log_2 10$ veces el número de dígitos decimales. El tamaño de la clave debe ser evaluada con el paso de los años, ya que durante ese tiempo puede haber surgido un algoritmo que factorice un número en tiempo reducido.

El criptosistema RSA no puede tener la misma velocidad que los criptosistemas convencionales debido a que es relativamente complicado implementar la multiplicación de enteros módulo n y a que la exponenciación requiere de una serie de multiplicaciones.

3.7.1. Firma digital con RSA

Supongamos que para evitar estar generando diferentes números $n = pq$ para cada usuario del sistema, podemos fijar n una vez y para todos, por lo que cada usuario i tiene clave pública (n, e_i) y clave privada (n, d_i). A primera vista parecer ser que esto funciona: si **B**ob le envía un texto cifrado $C \equiv P^{e_A} \bmod n$ a **A**le, sólo ella podrá descifrar C al utilizar su clave privada d_A. Sin embargo, esto es incorrecto, ya que por la Proposición 3.7.2 otro usuario **V** del sistema puede factorizar n con su propia clave pública e_V y clave privada d_V. Una vez que el usuario **V** factoriza n puede encontrar la clave privada d_A a partir de la clave pública e_A de **A**le. Por esta razón, el módulo n nunca debe ser utilizado por más de una entidad.

Una vez hecha esta observación veamos como firmar un mensaje P. Tenemos dos casos, el primero es cuando $n_A < n_B$; esto es, un usuario **A** firma P con su clave privada y después lo cifra con la clave pública de **B**, es decir:

$$F \equiv P^{d_A} \bmod n_A,$$
$$C \equiv F^{e_B} \bmod n_B.$$

Para verificar la autenticidad del mensaje cifrado C, **B**ob utiliza su clave privada d_B para descifrar C y verifica la procedencia del mensaje utilizando la clave pública e_A de **A**:

$$F \equiv C^{d_B} \bmod n_B,$$
$$P \equiv F^{e_A} \bmod n_A.$$

Este esquema le garantiza al usuario **B** que el mensaje proviene del usuario **A**; si no fuera así no podría descrifrarlo aplicando estas transformaciones.

En el caso de que $n_A > n_B$, ella primero calcula,

$$F \equiv P^{e_B} \bmod n_B,$$
$$C \equiv F^{d_A} \bmod n_A.$$

Para el proceso de verificación de la firma **B** realiza las mismas operaciones que se hicieron para el caso $n_A < n_B$ pero en orden inverso.

3.8. Massey-Omura para transferencia de mensajes.

Supongamos que todo mundo se pone de acuerdo sobre el campo finito $\mathbb{F}_q$ que van a utilizar, el cual es fijo y de conocimiento público. Cada usuario del sistema elige de manera secreta un número aleatorio e entre 0 y $q-1$ tal que $(e, q-1) = 1$ entonces por la Proposición 1.2.3 el número e tiene un inverso $d = e^{-1} \bmod q-1$; esto es, $ed \equiv 1 \bmod q-1$ y lo calcula utilizando el algoritmo extendido de Euclides (Sección 1.1.1).

Si **A**le le quiere enviar un mensaje P a **B**ob, primero le envía

$$P^{e_A}.$$

Esto no le dice nada a **B**ob porque no puede recuperar P sin el conocimiento de d_A. Pero sin tratar de entenderlo, él lo eleva a su e_B y le regresa a **A**le

$$P^{e_A e_B}.$$

Ahora el tercer paso lo realiza **A**le al elevar $P^{e_A e_B}$ a la potencia d_A y se lo regresa a **B**ob (por el Teorema 1.1.18 tenemos $P^{d_A e_A} = P^{1+t(q-1)} = P(P^{q-1})^t = P \cdot 1^t = P$, de donde se sigue que **A**le le está regresando P^{e_B}), quien ahora puede leer el mensaje al elevarlo a la potencia d_B.

La idea de este sistema es simple y puede generalizarse en situaciones donde se estén utilizando otros procesos además de la exponenciación en campos finitos. Sin embargo, notamos que es necesario utilizar un buen esquema de firma junto con el sistema Massey-Omura. De lo contrario, cualquier persona **C** que intersecte P^{e_A} podría hacerse pasar por **B**, regresándole $P^{e_A e_C}$ a **A**le; sin pensar que el intruso estaba usando su propio e_C, ella lo eleva a su d_A permitiéndole a **C** que pueda leer el mensaje. Así que el mensaje $P^{e_A e_B}$ de **B**ob a **A**le debe estar acompañado por alguna autenticación, es decir, algún mensaje en algún esquema de firma con la que sólo **B**ob puede enviarlo.

También notamos que después de que un usuario tal como **B** ó **C** han descifrado varios mensajes P y de que conocen varios pares (P, P^{e_A}), ellos no pueden usar esa información para determinar e_A.

3.9. El criptosistema ElGamal

Este criptosistema está basado en la dificultad de calcular logaritmos discretos sobre campos finitos y es en esencia una variante del sistema de distribución de clave Diffie-Hellman (Sección 3.4).

Iniciamos con un campo finito muy grande $\mathbb{F}_q$ y un elemento $g \in \mathbb{F}_q^*$ (de preferencia, pero no necesariamente, un generador). Supongamos que nuestras unidades de mensaje de texto en claro P tienen equivalentes numéricos en $\mathbb{F}_q$, es decir $P \approx \mathbb{F}_q$. Cada usuario

A elige aleatoriamente un entero a_A en el rango $0 < a_A < q-1$. Este entero a_A es la clave privada, mientras que la clave pública de cifrado es el elemento $g^{a_A} \in \mathbb{F}_q$.

Para enviarle a **A**le un mensaje P, elegimos un entero aleatorio k y le enviamos el siguiente par de elementos de $\mathbb{F}_q$:

$$(g^k, Pg^{a_A k}).$$

Podemos calcular $g^{a_A k}$ sin conocer a_A : simplemente elevamos g^{a_A} a la potencia $k-$ *ésima*. Ahora **A** puede recuperar P de este par al elevar el primer elemento g^k a la a_A-ésima potencia y dividiendo el segundo elemento por $g^{a_A k}$ (ó equivalentemente, elevar g^k a la $(q-1-a)-$ *ésima* potencia y multiplicarla por el segundo elemento $Pg^{a_A k}$, por el Teorema 1.1.18 tenemos que $g^{k(q-1-a_A)} Pg^{a_A k} = g^{k(q-1)} P = P$).

El proceso de descifrado permite recuperar el texto claro porque

$$g^{-a_A k} Pg^{a_A k} \equiv P \bmod q.$$

En otras palabras, lo que enviamos a **A** consiste de un disfraz del mensaje –P está "usando una máscara" g^{ak}– junto con una pista, a saber g^k, que puede usarse para quitar la máscara (pero la pista puede usarse sólo por alguien que conoce a_A).

Alguien quien pueda resolver el problema del logaritmo discreto en $\mathbb{F}_q$ rompe el criptosistema al encontrar la clave secreta de descifrado a de la clave pública de descifrado g^a. En teoría, podría haber una forma para encontrar g^{ak} a partir del conocimiento de g^k y g^a (y por lo tanto romper el cifrado) sin resolver el problema del logaritmo discreto. Sin embargo como mencionamos en la Sección 3.4.1 se conjetura que no hay forma de obtener g^{ak} a partir del conocimiento de g^a, g^k sin resolver esencialmente el problema de logaritmo discreto.

Ejemplo 3.9.1 *Sea $q = 1777$ y $g = 6$. **B** elige la clave privada $b = 1009$ y calcula*

$$g^b = 6^{1009} \equiv 1729 \ mod\ 1777.$$

*La clave pública de **B** es $(p, g, g^b) = (1777, 6, 1729)$, la cual **A** obtiene de una autoridad certificadora. Para cifrar el mensaje $m = 1483$, **A** selecciona aleatoriamente $k = 701$ y calcula*

$$\beta = 6^{701} \equiv 1664 \ mod\ 1777,$$

$$\gamma \equiv 1483 * 1729^{701} \equiv 1241 \ mod\ 1777.$$

*Entonces ella envía $(\beta, \gamma) = (1664, 1241)$ a **B**. Para descifrar, **B** calcula*

$$\beta^{q-1-b} = 1664^{1777-1-1009} \equiv 1572 \ mod\ 1777,$$

y recupera m al calcular

$$m = 1572 \cdot 1241 \ mod\ 1777 = 1483.$$

3.9.1. Firma digital ElGamal

Partimos de que a y g^a, las claves privada y pública respectivamente, son generadas de acuerdo al método descrito en el criptosistema ElGamal.

Para firmar un mensaje **A** elige un número entero aleatorio k tal que $k < p-1$ y además $(k, p-1) = 1$. Sea $h = H(m)$ el valor hash del mensaje m. **A** calcula

$$\begin{aligned} X &\equiv g^k \bmod p, \\ s &\equiv k^{-1}(h - aX) \bmod p-1. \end{aligned} \tag{3.2}$$

Entonces la firma de **A** es el par (X, s).

Para verificar la autenticidad del mensaje m, **B**ob debe obtener una copia auténtica de la clave pública g^a de **A.** Verifica que $1 \leq X \leq p-1$, si no ocurre así entonces rechaza la firma. Enseguida calcula

$$v_1 \equiv g^{aX} X^s \bmod p.$$

Luego calcula el valor hash h del mensaje m y además

$$v_2 \equiv g^h \bmod p.$$

Acepta la firma si y sólo si $v_1 = v_2$; en este caso m es un mensaje válido. Observamos que la etapa de la verificación de la firma funciona por la siguiente razón: al multiplicar ambos lados de (3.2) por k tenemos que:

$$ks \equiv h - aX \bmod p-1,$$

$$h \equiv ks + aX \bmod p-1.$$

La congruencia anterior y el Corolario ( ) implican que $g^h \equiv g^{ks+aX} \bmod p$, por lo tanto

$$g^h \equiv g^{ks} g^{aX} \equiv g^{aX} X^s \bmod p.$$

De esta manera vemos que $v_2 = v_1$.

3.10. DSS (Digital Signature Standar)

En 1991 el NIST (National Institute of Standards and Technology) propuso una firma digital estándar (DSS), la cual se basa en el problema del logaritmo discreto en un campo finito y es una variante de la firma digital ElGamal. Veamos como funciona el DSS.

Para iniciar el esquema (a fin de que posteriormente sea capaz de firmar mensajes) cada usuario **A** hace lo siguiente:

1. Elige un número primo q de alrededor de 160 bits y luego otro primo p de 1024 bits tal que $p \equiv 1 \bmod q$.

2. Elige un generador g del único subgrupo cíclico de $\mathbb{F}_p^*$ de orden q (elige un elemento aleatorio $g_0 \in \mathbb{F}_p^*$ y calcula $g = g_0^{(p-1)/q} \bmod p$, si el número obtenido es distinto de 1 entonces será un generador).

3. Toma un entero aleatorio x en el rango $0 < x < q$ como su clave secreta y hace su clave pública igual a $y = g^x \bmod p$.

Ahora supongamos que **A**le quiere firmar un mensaje m. Lo que hace primero es aplicar la función hash estándar SHA-1 (ver [11]) a su texto en claro m, obteniendo un entero h en el rango $0 < h < q$. Enseguida elige un entero aleatorio k en el rango $1 \leq k \leq q-1$, calcula

$$X \equiv g^k \bmod p$$

y pone

$$r \equiv X \bmod q$$

(es decir, primero se calcula g^k módulo p y el resultado X se reduce módulo q), si $r = 0$ entonces selecciona otro k. Finalmente, **A** encuentra un entero s tal que

$$sk \equiv (h + xr) \bmod q,$$

es decir

$$s \equiv k^{-1}(h + xr) \bmod q,$$

pero si $s = 0$ entonces busca otra k. La firma de **A** para el mensaje m es el par (r, s) de enteros módulo q.

Entonces cuando **A**le quiera enviarle un mensaje firmado a **B**ob, ella tiene que enviarle $(m, (r, s))$.

Para verificar la firma de **A**le, **B**ob comprueba que r, s sean enteros en el intervalo $[1, q-1]$, calcula el valor hash h del mensaje m y

$$\begin{aligned} u_1 &= s^{-1}h \bmod q, \\ u_2 &= s^{-1}r \bmod q. \end{aligned}$$

Luego, **B**ob calcula

$$\begin{aligned} X &= g^{u_1}g^{u_2} \bmod p, \\ v &= X \bmod q. \end{aligned}$$

Él acepta la firma si y sólo si $v = r$. Notamos que $g^{u_1}g^{u_2} = g^{s^{-1}(h+xr)} = g^k \bmod p$.

La seguridad del DSS se basa en dos problemas de logaritmo discreto distintos pero relacionados. Uno es el problema del logaritmo discreto en $\mathbb{F}_p^*$ y el otro es problema del logaritmo discreto en el subgrupo cíclico de orden q en $\mathbb{F}_p^*$. Para un p grande (1024 bits),

el mejor algoritmo conocido para este problema es el método de Pollard y toma $\sqrt{\pi q/2}$ pasos. Si $q \approx 2^{160}$, entonces la expresión anterior representa una cantidad no factible de cálculos, debido a esto el DSA no es vulnerable a su ataque. Además este esquema tiene la ventaja de que las firmas son bastante cortas (consiste de un par de números de 160 bits) facilitando su almacenamiento.

3.11. El sistema Merkle-Hellman

En esta sección introducimos otro tipo de criptosistema de clave pública, el cual se basa en el "*problema de la mochila*". Supongamos que tenemos una mochila de volumen V y un número k de cosas de volumen $v_i, i = 0, \ldots k-1$ para guardarlas en dicha mochila. Lo que se quiere lograr es desperdiciar el menor espacio posible, para hacerlo se necesita encontrar un subconjunto de las k cosas que llenen exactamente la mochila. En otras palabras, necesitamos encontrar un subconjunto $I \subset \{0, \ldots, k-1\}$ tal que $\sum_{i \in I} v_i = V$, si tal conjunto I existe. Además asumimos que V y los v_i's son enteros positivos. Una forma equivalente para expresar el problema de la mochila es como sigue:

Dado un conjunto $\{v_i\}$ de k enteros positivos y un entero V, encontrar un entero $n = (\epsilon_{k-1}\epsilon_{k-2} \ldots \epsilon_1\epsilon_0)_2$ de k bits (donde $\epsilon_i \in \{0, 1\}$ son los dígitos binarios de n) tal que

$$\sum_{i=0}^{k-1} \epsilon_i v_i = V,$$

si tal n existe.

Notar que puede que no haya ninguna solución n, muchas soluciones ó que n sea la única solución, dependiendo de la k-tupla (v_i) y el entero V.

El *problema de la mochila supercreciente* es un caso especial del problema de la mochila. Este es el caso cuando los v_i's, arreglados de manera creciente $v_0 < v_1 < \ldots v_{k-1}$, tienen la propiedad

$$v_i > \sum_{j=0}^{i-1} v_j,$$

para cada $i \in \{1, 2, \ldots, k-1\}$; es decir, cada v_i es mayor que la suma de todos los v_i anteriores.

Es conocido que el problema de la mochila es un problema **NP**-completo (ver [28, pag. 118]). Es decir, no se conoce ningún algoritmo de tiempo polinomial que resuelva el problema de la mochila.

Sin embargo, el problema de la mochila supercreciente es mucho más fácil de resolver que el problema general de la mochila. A saber, iniciamos con el v_i más grande tal que

$v_i \leq V$. Incluimos el correspondiente i en nuestro subconjunto I (es decir, tomamos $\epsilon_i = 1$, en caso contrario $\epsilon_i = 0$), reemplazamos V por $V - v_i$, continuamos en forma descendente hasta encontrar aquél v_i que sea menor o igual a esta diferencia.

Ejemplo 3.11.1 *Consideramos la secuencia supercreciente* $(2, 3, 7, 15, 31)$ *y* $V = 24$. *Entonces, trabajando de derecha a izquierda vemos que* $\epsilon_4 = 0$, $\epsilon_3 = 1$ (*sustituimos* 24 *por* $24 - 15 = 9$), $\epsilon_2 = 1$ (*sutituimos* 9 *por* $9 - 7 = 2$), $\epsilon_1 = 0$ (*ya que* $3 > 2$), $\epsilon_0 = 1$ (*ya que* $2 = 2$). *Por lo que* $n = (01101)_2 = 13$.

Ahora construimos el criptosistema Merkle-Hellman (también llamado el sistema de la Mochila). Supongamos que nuestras unidades de texto claro P tienen enteros de k-bits como sus equivalentes numéricos. Por ejemplo, si nuestras unidades de mensaje son caracteres en un alfabeto de 27 letras, entonces a cada letra le corresponde un entero de 5-bits, es decir del $0 = (00000)_2$ al $26 = (11010)_2$.

Para la generación de claves un usuario elige una k-tupla supercreciente $\{v_0, \ldots, v_{k-1}\}$, un entero m tal que $m > \sum_{i=0}^{k-1} v_i$, y un entero a tal que $0 < a < m$ y $(a, m) = 1$. Esto se hace por medio de un proceso aleatorio. Después calcula $b = a^{-1} \bmod m$ y también calcula la $k-tupla$ $\{w_i\}$ definida por $w_i = av_i \bmod m$. El usuario mantiene en secreto los números v_i, m, a, b y publica la $k-tupla$ $\{w_i\}$. Esto es, la clave de cifrado es $k_E = \{w_0, \ldots, w_{k-1}\}$ y la clave de descifrado es $k_D = (b, m)$ (junto con la clave de cifrado, este par ayuda a determinar $\{v_0, \ldots, v_{k-1}\}$).

Alguien que quiera enviar un mensaje de texto claro de $k-bits$ $P = (\epsilon_{k-1}\epsilon_{k-2} \ldots \epsilon_1\epsilon_0)_2$ a un usuario con clave de cifrado $\{w_i\}$, calcula

$$C = f(P) = \sum_{i=0}^{k-1} \epsilon_i w_i$$

y transmite ese entero.

Para leer el mensaje, el usuario primero encuentra el mínimo residuo positivo V de $bC \bmod m$. Dado que $bC \equiv \sum_{i=0}^{k-1} \epsilon_i b w_i \equiv \sum_{i=0}^{k-1} \epsilon_i v_i$ (porque $bw_i \equiv bav_i \equiv v_i \bmod m$), se tiene que $V = \sum \epsilon_i v_i$ (donde se usa el hecho de que $V < m$ y $\sum \epsilon_i v_i \leq \sum v_i < m$ para convertir las congruencias módulo m en igualdad). Luego se usa el algoritmo del problema de la mochila supercreciente, para encontrar la única solución $P = (\epsilon_{k-1} \ldots \epsilon_0)_2$ al problema de encontrar un subconjunto de $\{v_i\}$ tal que su suma sea exactamente V. De esta forma se recupera P.

Un intruso quien sólo conoce $\{w_i\}$ se enfrenta con el problema de la mochila $C = \sum \epsilon_i w_i$ que no es un problema supercreciente, porque la propiedad supercreciente de la k-tupla de v_i se destruye cuando v_i se sustituye por el mínimo residuo no negativo de av_i módulo m. Así, el algoritmo no puede ser usado y una persona no autorizada parece encontrarse con un problema mucho más complicado.

Ejemplo 3.11.2 *Supongamos que nuestras unidades de texto claro son de una sola letra con equivalentes numéricos de 5 bits de* $(00000)_2$ *a* $(11010)_2$. *Supongamos que nuestra clave secreta de descifrado es la secuencia supercreciente* $(2, 3, 7, 15, 31)$. *Elegimos* $m = 61$, $a = 17$; *entonces* $b = 18$ *y la clave de cifrado es* $(34, 51, 58, 11, 39)$. *Para enviar el mensaje 'HOLA' calculamos*

$$\begin{aligned} H &= (00111)_2 \longmapsto 34 + 51 + 58 = 143, \\ O &= (01111)_2 \longmapsto 34 + 51 + 58 + 11 = 154, \\ L &= (01011)_2 \longmapsto 34 + 51 + 11 = 96, \\ A &= (00000) \longmapsto 0. \end{aligned}$$

Para leer el mensaje $(143, 154, 96, 0)$ multiplicamos cada número por 18 y reducimos módulo 61 obteniendo $12, 27, 20, 0$. Procediendo como en el ejemplo 3.11.1 con $V = 12$, $V = 27$, $V = 20$, $V = 0$ recuperamos el texto claro (00111), $(01111)_2$, $(01011)_2$, $(00000)_2$.

Por supuesto, como de costumbre no hay seguridad al usar unidades de mensaje de un sólo carácter con semejante valor pequeño de $k = 5$.

Debido a que el problema de romper el criptosistema pertenece a una clase de problema muy difícil de resolver (problema NP-completo), muchas personas pensaron que el sistema sería seguro.

Sin embargo habia un error en ése razonamiento. El tipo de problema de la mochila $C = \sum \epsilon_i w_i$ que debe resolverse, que no es un problema de mochila supercreciente, es no obstante un tipo muy especial; es decir, es obtenido de un problema supercreciente por una simple transformación, es decir, multiplicar todo por a y reducir módulo m.

En 1982, Shamir encontró un algoritmo polinomial en k para resolver este tipo de problema de la mochila. De esta manera, el cryptosistema Merkle-Hellman original no puede ser considerado como un criptosistema de clave pública seguro; en [28, pag. 302] se describe el algoritmo de tiempo polinomial para romper el sistema Merkle-Hellman.

3.12. Protocolo de conocimiento nulo

El *conocimiento nulo* (zero knowledge) es el nombre asignado a un concepto criptográfico desarrollado a principios de la década de los 80's, que trata el siguiente problema: Supongamos que alguien quiere probar que ha resuelto como hacer "algo" —encontrar la solución a una ecuación, probar un teorema, etc.— mientras que al mismo tiempo no desea comunicar ningún hecho acerca de la prueba ó solución.

El *probador*, a quién llamamos Pícara, es la persona que tiene la solución; el *verificador* Vivales es quien finalmente debe convencerse de que Pícara tiene una solución, pero sin tener la más remota idea de cuál es la solución.

Las pruebas de conocimiento nulo son una herramienta importante en la criptografía de clave pública, que han permitido plantear y resolver problemas. Algunos de estos

problemas van desde la simple autenticación de usuarios hasta sistemas complejos de votación electrónica y sistemas de pago digital.

3.12.1. Prueba de conocimiento nulo de haber encontrado un logaritmo discreto

Supongamos que $\mathcal{G}$ es un grupo finito de N elementos (cuya operación de grupo se escribirá multiplicativamene), b es un elemento fijo de $\mathcal{G}$ y y es un elemento de $\mathcal{G}$ para el cual Pícara ha encontrado un logaritmo discreto a la base b; es decir, ella ha resuelto la ecuación $b^x = y$ donde x es un entero positivo.

Ella quiere convencer a Vivales de que conoce x pero sin darle ninguna pista de cuál es el valor de x. Los pasos que realizan ambos son:

1. Pícara genera un entero aleatorio $e \in \mathbb{Z}^+$ tal que $e < N$ y le envía a Vivales $b' = b^e$.

2. Vivales lanza una moneda. Si cae sol, Pícara debe revelar e y Vivales verifica que b' es b^e.

3. Si cae águila, entonces Pícara debe revelar $r \equiv (x + e) \bmod N$, a tal punto que Vivales pueda verificar que $yb' = b^{x+e} = b^r$.

4. Los pasos 1. – 3. se repiten hasta que Vivales esté convencido de que Pícara conoce el valor de x del logaritmo discreto.

Notamos que si Pícara no conoce el valor de x (es decir, está tratando de engañar a Vivales), entonces no podrá responder a más de un posible resultado del lanzamiento de la moneda. Esto es, después de haber realizado el paso 1. ella puede responder si cae sol –pero no a las águilas– sin conocer x. Por otro lado, si ella cree que va a caer águila entonces le envía $b' = y^{-1}b^e$ (así que en el paso 3. ella puede enviarle e en lugar de $x+e$), pero estará en apuros si cae sol porque no conoce la potencia de b que da como resultado b', o sea $b^h = y^{-1}b^e = b'$ y esto es equivalente a resolver el problema del logaritmo discreto; de esta manera él se daría cuenta de que lo está engañando.

Ahora supongamos que **C**arlos no conoce $\log_b y$ pero puede adivinar qué es lo que va a caer al lanzar la moneda. Entonces **C**arlos puede simular los mismo pasos que Pícara dándole a Vivales información indistinguible de la que Pícara le habría dado a Vivales. Carlos no puede darle a Vivales ninguna información útil para encontrar el logaritmo discreto ya que ni él mismo tiene idea de cuál es el logaritmo discreto.

3.13. Transferencia inconsciente

Un *canal de transferencia inconsciente* de Pícara a Vivales es un sistema en el que Pícara le envía a Vivales dos paquetes de información cifrados y que verifica las siguientes condiciones:

1. Vivales puede descifrar y leer exactamente uno de los dos paquetes,
2. Pícara no sabe cuál de los dos paquetes puede leer Vivales y
3. tanto Pícara como Vivales están conscientes de las condiciones 1. y 2.

A primera vista esto podría parecer un poco extraño. Sin embargo tal canal resulta ser un concepto fundamental en criptografía. A continuación describimos una forma de obtener un canal de transferencia inconsciente basado en la intratabilidad del problema del logaritmo discreto.

Más precisamente, supongamos que tenemos un campo finito $\mathbb{F}_q$ y un elemento fijo $b \in \mathbb{F}_q^*$ tal que dados b^x y b^y es difícil encontrar b^{xy} (ver sección 3.4.1).

Además supongamos que es fácil calcular el mapeo $\psi : \mathbb{F}_q \to \mathbb{F}_2^n$ y que la imagen de este mapeo contiene a F_2^{n-1} (todas las n-tuplas cuyo último bit es 0). Por ejemplo, si q es un primo p entonces podemos elegir n de manera que $2^{n-1} < p < 2^n$ y mapear cualquier elemento de $\mathbb{F}_p$ a su sucesión de dígitos binarios.

Supongamos que las unidades de mensaje también son n-tuplas de bits, es decir, elementos $m \in \mathbb{F}_2^n$. Finalmente supongamos que un elemento $C \in \mathbb{F}_q^*$, fijo y para todos, se ha elegido de tal manera que nadie conoce su logaritmo discreto $log_b C$. Este elemento podría haber sido otorgado por un "centro confiable" ó creado por un método aleatorio.

La transferencia inconsciente funciona de la siguiente manera: Vivales elige un entero aleatorio $x, 0 < x < q-1$, y también un elemento aleatorio $i \in \{1,2\}$. Vivales hace $\beta_i = b^x$ y $\beta_{3-i} = C/b^x$, luego coloca en un directorio público su "clave pública" (β_1, β_2) y mantiene en secreto los enteros x e i. Notamos que él no conoce el logaritmo discreto de β_{3-i} –que denotamos por x'– ya que si lo conociera entonces él encontraría el logaritmo discreto de $C = \beta_i \beta_{3-i}$, contrario a la suposición que se hizo anteriormente.

Ahora supongamos que Pícara tiene una unidad de mensaje $m_1 \in \mathbb{F}_2^n$ del primer paquete y una unidad de mensaje $m_2 \in \mathbb{F}_2^n$ del segundo paquete. Selecciona dos enteros aleatorios $0 < y_1, y_2 < q-1$, le envía a Vivales los siguientes elementos de $\mathbb{F}_q^*$ y de $\mathbb{F}_2^n$ respectivamente:

$$\begin{array}{rcl} & & b^{y_1},\ b^{y_2}, \\ \alpha_1 & = & m_1 + \psi(\beta_1^{y_1}),\ \alpha_2 = m_2 + \psi(\beta_2^{y_2}). \end{array}$$

Pícara mantiene en secreto y_1 y y_2.

Como $\beta_i^{y_i} = (b^{y_i})^x$ y Vivales conoce b^{y_i}, x él puede determinar $\psi(\beta_i^{y_i})$ y $m_i = \alpha_i + \psi(\beta_i^{y_i})$. Sin embargo, si él quiere quiere encontrar m_{3-i}, tendrá que calcular $\beta_{3-1}^{y_{3-i}} = b^{x' y_{3-i}}$ conociendo únicamente $b^{y_{3-i}}$ y $b^{x'}$ pero no y_{3-i} ó x'. Lo cual es imposible, por la suposición de Diffie-Hellman.

Pícara puede verificar fácilmente que $\beta_1\beta_2 = C$ y de ésta manera asegurararse que Vivales no conoce el logaritmo discreto de ambos elementos de su clave pública (β_1, β_2). Dado que el interés de Vivales es obtener la mayor información posible, Pícara debe asegurarse que él conozca el logaritmo discreto de uno de los dos elementos. Pero no hay forma de que Pícara pueda distinguir entre β_1 y β_2 con el propósito de determinar cuál obtuvo Vivales como b^x y cuál como C/b^x. De esta manera, tanto Vivales como Pícara pueden asegurarse de que las condiciones 1. y 2. se cumplen.

Capítulo 4

Curvas elípticas

Las curvas elípticas han sido objeto de estudio en diferentes áreas de las matemáticas como la variable compleja, la teoría de números, la geometría algebraica. Existe una inmensa cantidad de literatura sobre el tema, algunas referencias son Silverman [40], [41], Koblitz [22], Housemöller [13] y Brieskorn [2]. Las curvas elípticas también figuran en la reciente prueba del Último Teorema de Fermat por Adrew Wiles. Originalmente su estudio fue puramente teórico, pero recientemente se utilizan para idear algoritmos de factori-zación de enteros, pruebas de primalidad y para la criptografía de clave pública.

Debemos notar que aunque frecuentemente trabajamos sobre campos finitos escribimos " $=$ " en lugar de " $\equiv$ ", utilizando la misma notación de los libros [17], [18].

4.1. El plano proyectivo

Definición 4.1.1 *Sea $\mathbb{K}$ un campo. Denotamos por $\mathbb{K}[x_1, \ldots x_n]$ el conjunto de polinomios en las variables $x_1, \ldots x_n$ con coeficientes en $\mathbb{K}$.*

Definición 4.1.2 *El "grado total" de un monomio $x^i y^j$ es $i+j$ y el "grado total" de un polinomio $F \in \mathbb{K}[x,y]$ es el máximo grado total de los monomios en F con coeficientes no cero. Si F tiene grado total n, definimos su correspondiente polinomio homogéneo $\widetilde{F} \in \mathbb{K}[x,y,z]$ como sigue:*

$$\widetilde{F}(x,y,z) = z^n F\left(\frac{x}{z}, \frac{y}{z}\right).$$

Ejemplo 4.1.3 *Si $F(x,y) = y^2 - (x^3 - x)$, tenemos que $\widetilde{F}(x,y,z) = y^2 z - x^3 + xz^2$.*

Comentario 4.1.4 *Notamos que $F(x,y) = \widetilde{F}(x,y,1)$.*

Supongamos que $F \in \mathbb{K}[x,y]$ tiene grado total n, consideremos ahora las tripletas $x, y, z \in \mathbb{K}$ tal que $\widetilde{F}(x,y,z) = 0$. Notamos que:

1. para cualquier $0 \neq \lambda \in \mathbb{K}$, $\widetilde{F}(\lambda x, \lambda y, \lambda z) = \lambda^n \widetilde{F}(x, y, z)$, (homogeneidad);
2. para cualquier $0 \neq \lambda \in \mathbb{K}$, $\widetilde{F}(\lambda x, \lambda y, \lambda z) = 0$ si y sólo si $\widetilde{F}(x, y, z) = 0$.

En particular, para $z \neq 0$ tenemos $\widetilde{F}(x, y, z) = 0$ si y sólo si $F(\frac{x}{z}, \frac{y}{z}) = 0$.

Definimos una relación de equivalencia en $\mathbb{K}^3 \setminus (0,0,0)$ como sigue:

$$(x, y, z) \sim (x', y', z')$$

si existe $0 \neq \lambda \in \mathbb{K}$ tal que $(x', y', z') = \lambda(x, y, z)$.

Intuitivamente, una clase de equivalencia $[(x, y, z)]$ es una recta en $\mathbb{K}^3 \setminus (0,0,0)$ que pasa por el origen con dirección (x, y, z) y se le llama punto proyectivo.

Debido a 2., si $(x, y, z) \in \mathbb{K}^3 \setminus (0,0,0)$ es una solución de la ecuación $\widetilde{F}(x, y, z) = 0$ entonces todo elemento en $[(x, y, z)]$ también es una solución. Por lo tanto, para encontrar las soluciones de $\widetilde{F} = 0$ basta verificar en un sólo representante de cada clase de equivalencia.

Definición 4.1.5 *El plano proyectivo* $\mathbb{P}^2_{\mathbb{K}}$ *es el conjunto:*

$$\mathbb{P}^2_{\mathbb{K}} = \left\{ [(x, y, z)] : (x, y, z) \in \mathbb{K}^3 \setminus (0,0,0) \right\}.$$

Una manera de visualizar $\mathbb{P}^2_{\mathbb{K}}$ es como sigue: consideremos el plano $z = 1$, claramente todas las rectas en $\mathbb{K}^3 \setminus (0,0,0)$ a través del origen, excepto aquellas que están en el plano xy, tienen un único punto de intersección con este plano. Esto es, cada clase de equivalencia $[(x, y, z)]$ con coordenada $z \neq 0$ contiene una única tripleta de la forma $(x, y, 1)$. Así, pensamos tales clases de equivalencia como puntos en el plano xy (Ver la figura 1).

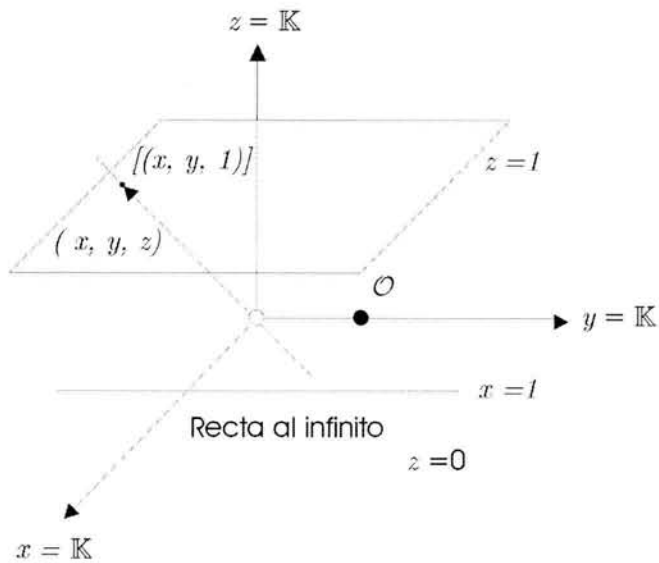

Figura 1: Plano proyectivo.

Por otro lado, las clases de equivalencia de la forma $[(x, y, 0)]$ forman la "*recta al infinito*". Dado que toda recta en el plano xy a través del origen (excepto el eje Y) intersecta la recta $x = 1$ en un solo punto, identificamos la recta al infinito con la recta $x = 1$ junto con el *"punto al infinito"* $\mathcal{O}$=$[(0, 1, 0)]$) (ver la figura 2).

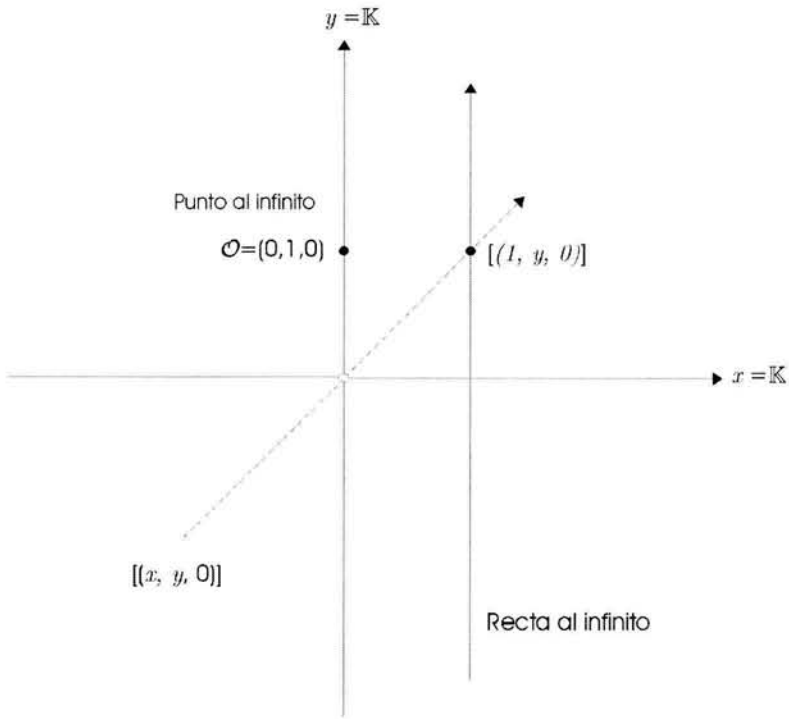

Figura 2: Recta al infinito.

Definición 4.1.6 *Dado un polinomio homogéneo* $\widetilde{F} \in \mathbb{K}[x, y, z]$, *el conjunto solución de* $\widetilde{F}$ *consiste de los puntos proyectivos* $[(x, y, z)]$ *en* $\mathbb{P}^2_{\mathbb{K}}$ *tal que* $\widetilde{F}(x, y, z) = 0$.

Comentario 4.1.7 *Los puntos* $[(x, y, z)]$ *del conjunto solución con* $z \neq 0$ *son los puntos* $[(x, y, 1)]$ *para los cuales* $\widetilde{F}(x, y, z) = F(x, y) = 0$. *Los puntos restantes del conjunto solución están en la recta al infinito.*

El conjunto solución de $\widetilde{F}(x, y, z) = 0$ se llama la "*completación proyectiva*" de la curva $F[x, y] = 0$. De ahora en adelante, cuando hablemos de una "recta", una "sección cónica", una "curva elíptica", etc., se le pensará en el plano proyectivo $P^2_{\mathbb{K}}$.

Ejemplo 4.1.8 *Sea* $F(x, y) = y^2 - x^3 + x$, $\widetilde{F}(x, y, z) = y^2z - x^3 + xz^2$. *Si* $[x, y, 0]$ *es un punto en la recta al infinito que satisface* $0 = \widetilde{F}(x, y, 0) = -x^3$, *entonces* $[x, y, 0] = [0, 1, 0] = \mathcal{O}$.

4.2. Ecuación de Weierstrass

Definición 4.2.1 *Una ecuación de Weierstrass es una ecuación homogénea de grado tres de la forma:*

$$Y^2Z + a_1XYZ + a_3YZ^2 = X^3 + a_2X^2Z + a_4XZ^2 + a_6Z^3,$$

donde $a_1, a_2, a_3, a_4, a_6 \in \mathbb{K}$.

Definición 4.2.2 *La ecuación de Weierstrass es suave o no singular si para todo* $P = [(x, y, z)] \in P^2_{\mathbb{K}}$ *que satisface:*

$$\widetilde{F}(X, Y, Z) = Y^2Z + a_1XYZ + a_3YZ^2 - X^3 - a_2X^2Z - a_4XZ^2 - a_6Z^3 = 0, \quad (4.1)$$

al menos una de las tres derivadas parciales $\frac{\partial F}{\partial X}$, $\frac{\partial F}{\partial Y}$, $\frac{\partial F}{\partial X}$ *es distinta de cero en* P *(ver [27]).*

Comentario 4.2.3 *Si las tres derivadas parciales se anulan en* P, *entonces a* P *se le conoce como punto singular y entonces se dice que la ecuación de Weierstrass es singular.*

Definición 4.2.4 *Una curva elíptica* E *sobre un campo* $\mathbb{K}$ *es el conjunto solución en* $\mathbb{P}^2_{\mathbb{K}}$ *de una ecuación de Weierstrass suave.*

Comentario 4.2.5 *Si* $[(1, Y, 0)]$ *es un punto en la recta al infinito de* $P^2_{\mathbb{K}}$ *que satisface* $\widetilde{F}(1, Y, 0) = 0$ *entonces* $X^3 = 0$, *pero la única clase de equivalencia* $[(X, Y, Z)]$ *con* $X = Z = 0$ *es* $\mathcal{O} = [(0, 1, 0)]$, *es decir, el punto infinito.*

Por comodidad, escribimos la equación de Weierstrass para una curva elíptica usando coordenadas no homogéneas (plano afín) $x = \frac{X}{Z}, y = \frac{Y}{Z}$, y consideramos el conjunto solución de la ecuación

$$F(x, y) = y^2 + a_1xy + a_3y - x^3 - a_2x^2 - a_4x - a_6 = 0, \quad (4.2)$$

junto con el punto al infinito $\mathcal{O}$.

De la definición anterior y de los comentarios 4.1.7, 4.2.5 se sigue que una curva elíptica E es el conjunto solución de la función F en el plano afín $A^2_{\mathbb{K}} = \mathbb{K} \times \mathbb{K}$ junto con el punto al infinito $\mathcal{O}$ y la denotamos por $E(\mathbb{K})$.

Al número de puntos en una curva elíptica $E(\mathbb{K})$ lo denotamos por $\#E(\mathbb{K})$.

Ahora damos algunos parámetros de las curvas que satisfacen (4.2), que nos ayudarán a dar una clasificación de las curvas elípticas.

Definición 4.2.6 *Sea E una curva definida por la ecuación (4.2), a partir de la cual definimos los siguientes coeficientes (ver [27, pag. 19]):*

$$\begin{aligned} b_2 &= a_1^2 + 4a_2, \\ b_4 &= a_1a_3 + 2a_4, \\ b_6 &= a_3^2 + 4a_6, \\ b_8 &= a_1^2a_6 + 4a_2a_6 - a_1a_3a_4 + a_2a_3^2 - a_4^2, \\ c_4 &= b_2^2 - 24b_4, \\ c_6 &= -b_2^3 + 36b_2b_4 - 216b_6, \\ \Delta &= -b_2^2b_8 - 8b_4^3 - 27b_6^2 + 9b_2b_4b_6, \\ j &= \frac{c_4^3}{\Delta}, \\ w &= \frac{dx}{2y + a_1x + a_3} = \frac{dy}{3x^2 + 2a_2x + a_4 - a_1y}. \end{aligned} \tag{4.3}$$

Cuando la característica del campo es distinta de 2 y 3: $\Delta = (c_4^3 - c_6^2)/12^3$.

Definición 4.2.7 *El parámetro Δ es el discriminante de la ecuación de Weierstrass, mientras que $j(E)$ es el j-invariante de E si $\Delta \neq 0$. El parámetro w es el invariante diferencial asociado con la ecuación de Weierstrass.*

Los siguientes resultados explican la importancia de estos parámetros.

Teorema 4.2.8 *La ecuación de Weierstrass es suave si y sólo si $\Delta \neq 0$ (ver [40, pag. 50]).*

Definición 4.2.9 *Se dice que dos curvas elípticas son isomorfas si son isomorfas como variedades proyectivas. Brevemente, dos variedades proyectivas V_1, V_2 definidas sobre un campo $\mathbb{K}$ son isomorfas sobre $\mathbb{K}$ si existe los morfismos $\phi : V_1 \to V_2$ y $\psi : V_2 \to V_1$ (ϕ, ψ definidas en $\mathbb{K}$) tal que $\psi \circ \phi$ y $\phi \circ \psi$ son los mapeos identidad sobre V_1 y V_2 respectivamente (ver [27, pag. 16]).*

Proposición 4.2.10 *Dos curvas elípticas sobre $\mathbb{K}$, car $\mathbb{K} \neq 2, 3$ son isomorfas si y sólo si tienen el mismo j-invariante (ver [40, pag.51]).*

El siguiente resultado relaciona la noción de isomorfismo de curvas elípticas con los coeficientes de la ecuación (4.2) que define a las curvas.

Teorema 4.2.11 *Dos curvas elípticas E_1 y E_2 dadas por las ecuaciones:*

$$\begin{aligned} E_1 &: y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6, \\ E_2 &: y^2 + \bar{a}_1xy + \bar{a}_3y = x^3 + \bar{a}_2x^2 + \bar{a}_4x + \bar{a}_6, \end{aligned}$$

son isomorfas en $\mathbb{K}$, *denotado por* $E_1 \cong E_2$, *si y solo si existen* $u, r, s, t \in \mathbb{K}$, $u \neq 0$, *tal que el cambio de variables*

$$(x, y) \longrightarrow (u^2x + r, u^3y + u^2sx + t) \tag{4.4}$$

transforma la ecuación E_1 *en* E_2. *La relación de isomorfismo es una relación de equivalencia (ver [27, pag.16]).*

Al cambio de variables (4.4) se le conoce como *cambio de variables permisibles.*

A continuación vemos como la ecuación de Weierstrass (4.2) puede tomar distintas formas (dependiendo de la característica del campo $\mathbb{K}$) mediante cambios de variables permisibles; en cada caso se dan los parámetro antes definidos.

Teorema 4.2.12 *Sea* E *una curva elíptica definida por la ecuación (4.2). Existe una sustitución de la forma:*

$$(x, y) \longrightarrow (u^2x + r, u^3y + u^2sx + t) \qquad \text{con } u \in \mathbb{K}^* \text{ y } r, s, t \in \mathbb{K},$$

tal que E *es isomorfa a la curva* E' *con ecuación dada abajo. Distinguimos los siguientes casos:*

(a) si car $\mathbb{K} \neq 2, 3$ *el cambio de variable es* $(x, y) \longrightarrow (x - \frac{b_2}{12}, y - \frac{a_1x + a_3}{2})$,

$$y^2 = x^3 + a_4x + a_6, \qquad \Delta = -16(4a_4^3 + 27a_6^2), \qquad j = 12^3 \frac{4a_4^3}{4a_4^3 + 27a_6^2}. \tag{4.5}$$

(b) si car $\mathbb{K} = 3$ *y* $j(E) \neq 0$ *el cambio de variable es* $(x, y) \longrightarrow (x + \frac{a_4}{a_2}, y)$,

$$y^2 = x^3 + a_2x^2 + a_6, \qquad \Delta = -a_2^3, \qquad j = -\frac{a_2^3}{a_6}. \tag{4.6}$$

si car $\mathbb{K} = 3$ *y* $j(E) = 0$

$$y^2 = x^3 + a_4x + a_6, \qquad \Delta = -a_4^3, \qquad j = 0. \tag{4.7}$$

(c) si car $\mathbb{K} = 2$ *y* $j(E) \neq 0$ *el cambio de variable es* $(x, y) \longrightarrow (a_1^2x + \frac{a_3}{a_1}, a_1^3y + \frac{a_1^2a_4 + a_3^2}{a_1^3})$,

$$y^2 + xy = x^3 + a_2x^2 + a_6, \qquad \Delta = a_6, \qquad j = \frac{1}{a_6}. \tag{4.8}$$

si car $\mathbb{K} = 2$ *y* $j(E) = 0$ *el cambio de variable es* $(x, y) \longrightarrow (x + a_2, y)$,

$$y^2 + a_3y = x^3 + a_4x + a_6, \qquad \Delta = a_3^4, \qquad j = 0. \tag{4.9}$$

Prueba. (Ver [40, pag. 46])
Demostración de (a).

Si car $\mathbb{K} \neq 2$, completamos el trinomio cuadrado perfecto en (4.2):

$$(y + (\frac{a_1x + a_3}{2}))^2 = x^3 + a_2x^2 + a_4x + a_6 + (\frac{a_1x + a_3}{2})^2,$$

haciendo el cambio $(x, y) \longrightarrow (x, y - \frac{a_1x+a_3}{2})$ obtenemos la ecuación

$$y^2 = x^3 + \frac{1}{4}a_1^2x^2 + a_2x^2 + \frac{1}{2}a_1a_3x + a_4x + \frac{1}{4}a_3^2 + a_6$$

por (4.3) se convierte en:

$$y^2 = x^3 + \frac{b_2}{4}x^2 + \frac{b_4}{2}x + \frac{b_6}{4}.$$

Si además car $\mathbb{K} \neq 3$ en la ecuación anterior hacemos el cambio $(x, y) \longrightarrow (x - \frac{b_2}{12}, y)$ y obtenemos

$$\begin{aligned} y^2 &= x^3 - \frac{1}{48}b_2^2x + \frac{24}{48}b_4x + \frac{1}{864}b_2^3 - \frac{1}{24}b_4b_2 + \frac{b_6}{4}\ , \\ y^2 &= x^3 - \frac{c_4}{48}x - \frac{c_6}{864}, \end{aligned}$$

donde hemos usado (4.3), por lo tanto

$$y^2 = x^3 + \widetilde{a}_4x + \widetilde{a}_6$$

donde $c_4 = -48\widetilde{a}_4, c_6 = -864\widetilde{a}_6$.

Demostración de (c).

Si car $\mathbb{K} = 2$ entonces $b_2 = a_1^2$, $c_4 = b_2^2$, por lo tanto $j(E) = 0$ si y sólo si $a_1 = 0$. 1) Si $j(E) \neq 0$ entonces $a_1 \neq 0$. Haciendo el cambio $(x, y) \longrightarrow (a_1^2x + \frac{a_3}{a_1}, a_1^3y + \frac{a_1^2a_4+a_3^2}{a_1^3})$ en la ecuación (4.2) obtenemos:

$$\begin{aligned} &a_1^6y^2 + a_1^6xy + (a_1^2a_4 + a_3^2)x + \frac{a_1^4a_4^2 + a_3^4}{a_1^6} = \\ &a_1^6x^3 + (3a_1^3a_3 + a_1^4a_2)x^2 + (a_1^2a_4 + 3a_3^2)x + \frac{a_3^3}{a_1^3} + \frac{a_2a_3^2}{a_1^2} + \frac{a_4a_3}{a_1} + a_6, \end{aligned}$$

simplificamos

$$a_1^6y^2 + a_1^6xy = a_1^6x^3 + (3a_1^3a_3 + a_1^4a_2)x^2 + a_1^6a_6,$$

dividimos por a_1^6 y la ecuación se transforma en:

$$y^2 + xy = x^3 + \widetilde{a}_2x^2 + \widetilde{a}_6.$$

2) Si $j(E) = 0$ entonces $a_1 = 0$, haciendo el cambio $(x, y) \longrightarrow (x + a_2, y)$ en la ecuación (4.2) obtenemos:

$$y^2 + a_3 y = x^3 + 4a_2 x^2 + (5a_2^2 + a_4)x + 2a_2^3 + a_2 a_4 + a_6,$$

simplificamos y la ecuación toma la forma:

$$y^2 + a_3 y = x^3 + a_4 x + a_6.$$

■

A las curvas elípticas con ecuación (4.9) se les conoce como *curvas supersingulares.* Mientras que a las curvas elípticas con ecuación (4.8) se les conoce como *curvas no-supersingulares* .

La siguiente clase de curvas elípticas también conocidas como *curvas binarias anómalas* ó *curvas ABC*, fueron propuestas por primera vez para su uso criptográfico por Koblitz en [19].

Definición 4.2.13 *A las curvas elípticas del tipo*

$$\begin{aligned} y^2 + xy &= x^3 + 1 \\ y^2 + xy &= x^3 + x^2 + 1 \end{aligned} \tag{4.10}$$

se les conoce como curvas de Koblitz. De manera equivalente, una curva de Koblitz es aquella para la cual $\#E(\mathbb{F}_q) = q$.

Proposición 4.2.14 *Si una curva elíptica* E *está definida sobre el campo* $\mathbb{F}_p$ *entonces también está definida sobre* $\mathbb{F}_{p^n}$ *para* $n \geq 1$ *(ver [17, pag. 175]).*

4.3. Ley de Grupo

Una característica notable del conjunto de puntos en una curva elíptica es que se le puede dotar de una estructura de grupo abeliano. Para explicar como funciona geométricamente la operación de grupo, suponemos por el momento que la curva elíptica E está definida sobre el campo $\mathbb{K} = \mathbb{R}$.

Recordamos que la curva elíptica E consiste de los puntos $P(x, y)$ que satisfacen la ecuación (4.2) junto con el punto al infinito $\mathcal{O} = [0, 1, 0]$. Sea L una recta, como la ecuación (4.2) tiene grado tres se sigue que L intersectan a E en exactamente tres puntos, a saber P, Q y R.

Definición 4.3.1 *Sea* $P = (x, y) \in E$. *Definimos* $\ominus P = (x, -y)$.

Ahora definimos la ley de composición $\oplus$ sobre E.

Definición 4.3.2 *Sean $P, Q \in E$, L la recta que une los puntos P y Q (la recta tangente a E si $P = Q$) y R el tercer punto de intersección de L con E. Entonces definimos $P \oplus Q$ como $\ominus R$; es decir, el simétrico (con respecto al eje x) del tercer punto de intersección.*

En la figura 3 podemos visualizar geométricamente la suma de los puntos P y Q en una curva elíptica definida sobre el campo $\mathbb{R}$.

A continuación justificamos el uso del símbolo $\oplus$.

Proposición 4.3.3 *La ley de composición tiene las siguientes propiedades:*

1. *Si una recta L intersecta a E en los puntos (no necesariamente distintos) P, Q, R entonces*
$$(P \oplus Q) \oplus R = \mathcal{O}$$

2. $(P \oplus Q) \oplus R = P \oplus (Q \oplus R)$ *para todo $P, Q, R \in E$. Propiedad asociativa.*

3. $P \oplus \mathcal{O} = P$ *para todo $P \in E$ ($\mathcal{O}$ es el elemento neutro del grupo de puntos).*

4. *Sea $P = (x_1, y_1) \neq \mathcal{O}$. Hay un punto en E, denotado por $\ominus P$, tal que*
$$P \oplus (\ominus P) = \mathcal{O}.$$

5. $P \oplus Q = Q \oplus P$ *para todo P, $Q \in E$. Propiedad conmutativa.*

Teorema 4.3.4 $(E, \oplus)$ *es un grupo abeliano con $\mathcal{O}$ como neutro aditivo.*

La demostración se puede consultar en [40, pag. 55-57] ó en [18]. La parte difícil es la prueba de la propiedad asociativa. Notamos que $\ominus P$ es el elemento inverso de P.

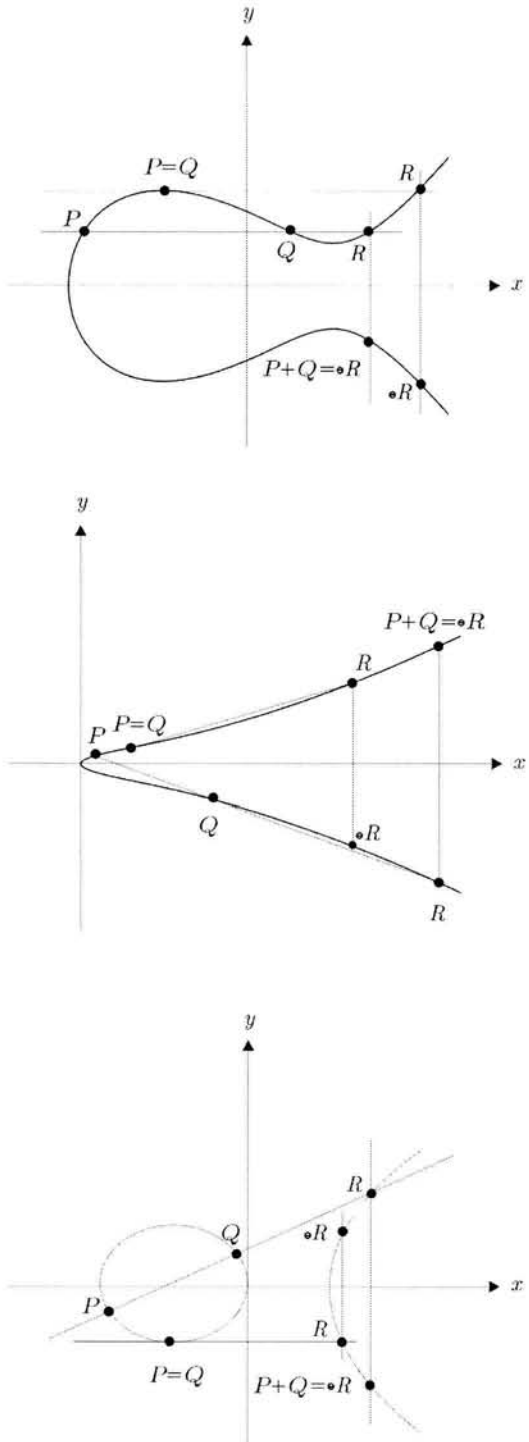

Figura 3: Descripción geométrica de la suma de puntos.

A continuación se dan las fórmulas explícitas para la operación de grupo en campos finitos. En [17, pag. 169] se muestra como se obtienen estas fórmulas a partir de la ecuación de Weierstrass (4.2).

Proposición 4.3.5 *Sea car* $\mathbb{K} \neq 2, 3$. *Si* $P = (x_1, y_1)$ *y* $Q = (x_2, y_2)$ *satisfacen la ecuación (4.5) entonces* $\ominus P = (x_1, -y_1)$. *Además, si* $Q \neq \ominus P$ *entonces* $P \oplus Q = (x_3, y_3)$ *donde:*

$$\begin{aligned} \lambda &= \begin{cases} (x_2 - x_1)^{-1}(y_2 - y_1) & si\ P \neq Q \\ (2y_1)^{-1}(3x_1^2 + a_4) & si\ P = Q \end{cases} \\ x_3 &= \lambda^2 - x_1 - x_2, \\ y_3 &= \lambda(x_1 - x_3) - y_1. \end{aligned}$$

Proposición 4.3.6 *Sea car* $\mathbb{K} = 2, j(E) \neq 0$. *Si* $P = (x_1, y_1)$ *y* $Q = (x_2, y_2)$ *satisfacen la ecuación (4.8) entonces* $\ominus P = (x_1, y_1 + x_1)$. *Además, si* $Q \neq \ominus P$ *entonces* $P \oplus Q = (x_3, y_3)$ *donde:*

$$\begin{aligned} \lambda &= (x_1 + x_2)^{-1}(y_1 + y_2) \quad si\ P \neq Q \\ x_3 &= \lambda^2 + \lambda + x_1 + x_2 + a_2, \\ y_3 &= \lambda(x_1 + x_3) + x_3 + y_1. \end{aligned}$$

$$\begin{aligned} \lambda &= x_1 + (x_1)^{-1} y_1 \quad si\ P = Q \\ x_3 &= x_1^2 + a_6 (x_1^2)^{-1}, \\ y_3 &= x_1^2 + (x_1 + y_1 x_1^{-1}) x_3 + x_3. \end{aligned}$$

Proposición 4.3.7 *Sea car* $\mathbb{K} = 2, j(E) = 0$. *Si* $P = (x_1, y_1)$ *y* $Q = (x_2, y_2)$ *satisfacen la ecuación (4.9) entonces* $\ominus P = (x_1, y_1 + a_3)$. *Además, si* $Q \neq \ominus P$ *entonces* $P \oplus Q = (x_3, y_3)$ *donde:*

$$\begin{aligned} \lambda &= (x_1 + x_2)^{-1}(y_1 + y_2) \quad si\ P \neq Q \\ x_3 &= \lambda^2 + x_1 + x_2, \\ y_3 &= \lambda(x_1 + x_3) + y_1 + a_3. \end{aligned}$$

$$\begin{aligned} \lambda &= (x_1 + x_2)^{-1}(y_1 + y_2) \quad si\ P = Q \\ x_3 &= \frac{x_1^4 + a_4^2}{a_3^2}, \\ y_3 &= (\frac{x_1^2 + a_4}{a_3})(x_1 + x_3) + y_1 + a_3. \end{aligned}$$

Para $n \in \mathbb{Z}$, $n > 0$ y $P \in E$, definimos $nP = \overbrace{P \oplus \ldots \oplus P}^{n \text{ veces}}$, $(-n)P = n(\ominus P)$ y $0P = \mathcal{O}$.

4.4. Propiedades de interés criptográfico

En criptografía no se considera nada práctico utilizar el campo $\mathbb{R}$ debido a errores de redondeo, por lo que restringimos nuestro estudio a las curvas elípticas definidas sobre los campos finitos $\mathbb{F}_q$, con $q = p^m$. Si $p > 3$ y car $\mathbb{F}_p \neq 2, 3$ entonces E está dada por la ecuación de la forma (4.5). Si $p = 2$ ó 3 entonces E está dada por las ecuaciónes (4.8, 4.9) ó (4.6, 4.7) respectivamente.

4.4.1. Estructura de grupo de los puntos de una curva elíptica

Sea E dada por la ecuación (4.2) definida sobre $\mathbb{F}_q$ con $q = p^m$, donde p es la característica de $\mathbb{F}_q$.

Al sustituir los diferentes valores de $x \in \mathbb{F}_q$ en la ecuación de Weierstrass 4.2, podemos observar que hay a lo más dos soluciones para cada $x \in \mathbb{F}_q$ (en particular lo podemos observar en el ejemplo 4.4.10 con la curva elíptica $y^2 = x^3 + x + 6$), de esta manera sabemos que $\#E(\mathbb{F}_q) \leq 2q + 1$, es decir, existen a lo más $2q$ soluciones más el punto al infinito $\mathcal{O}$. Dado que una ecuación cuadrática "aleatoriamente escogida" tiene una probabilidad de $1/2$ de ser resuelta en $\mathbb{F}_q$, uno esperaría que el orden de magnitud de puntos en la curva elíptica sea q, esto es $\#E(\mathbb{F}_q) \approx q$.

El siguiente resultado, conjeturado por E. Artin en su tésis y probado por Hasse en 1930, confirma que este razonamiento es correcto.

Teorema 4.4.1 *Sea E una curva elíptica sobre un campo finito $\mathbb{F}_q$. Entonces*

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

(ver [40, pag.131]).

El Teorema de Hasse nos da una cota para el número de puntos en una curva elíptica definida sobre $\mathbb{F}_q$. Más concretamente, Waterhouse probó en [45] lo siguiente,

Teorema 4.4.2 *Sea E una curva elíptica sobre el campo finito $\mathbb{F}_q$ y $N = \#E(\mathbb{F}_q)$ dado por $N = 1 + q - t$, donde t es un entero tal que $|t| \leq 2\sqrt{q}$. Entonces todos los posibles valores de N satisfacen las siguientes condiciones:*

1. *Si n es impar entonces $t = 0$,*
2. *Si n es impar y $p = 2$ ó 3 entonces $t = \pm p^{(n+1)/2}$,*
3. *Si n es par entonces $t = \pm 2\sqrt{q}$,*
4. *Si n es par y $3 \nmid p - 1$ entonces $t = \pm\sqrt{q}$,*
5. *Si n es par y $4 \nmid p - 1$ entonces $t = 0$.*

Ejemplo 4.4.3 *1. Si $n = 155$ y $p = 2$ entonces $\#E(\mathbb{F}_{2^{155}}) = 1 + 2^{155} \pm 2^{78}$ (caso 2.),*

2. Si $n = 10$, $p = 2$ entonces $\#E(\mathbb{F}_{2^{10}}) = 1 + 2^{10} \pm 2^5$ (caso 3.),

3. Si $n = 100$ y $p = 7$ entonces $\#E(\mathbb{F}_{7^{100}}) = 1 + 7^{100}$ (caso 5.).

Definición 4.4.4 *El orden de la curva elíptica $E(\mathbb{F}_q)$ es $N = \#E(\mathbb{F}_q)$.*

Una consecuencia del teorema de Hasse es que podemos elegir, de manera aleatoria y uniforme, un punto P sobre una curva elíptica $E(\mathbb{F}_q)$. Esto se realiza como sigue: Primero elegimos un elemento aleatorio $x_1 \in \mathbb{F}_q$. Si x_1 es la coordenada de algún punto en $E(\mathbb{F}_q)$ entonces al resolver la raíz cuadrada podemos encontrar y_1 tal que $(x_1, y_1) \in E(\mathbb{F}_q)$.

Del Teorema de Hasse, la probabilidad de que x_1 sea la coordenada de algún punto en $E(\mathbb{F}_q)$ es de al menos $1/2 - 1/\sqrt{q}$.

El concepto de orden puede aplicarse igualmente a los puntos de una curva elíptica.

Definición 4.4.5 *Sea $P \in E$. Si existe un entero positivo n tal que*

$$nP = \overbrace{P \oplus \ldots \oplus P}^{n \text{ veces}} = \mathcal{O},$$

entonces el orden de P es n y se dice que P es de orden finito. Si tal entero no existe entonces P es de orden infinito.

A nP lo denominamos como el múltiplo $n - ésimo$ de P.

El siguiente teorema nos proporciona el *tipo* de $E(\mathbb{F}_q)$. Usamos $\mathbb{Z}_n$ para denotar al grupo cíclico de n elementos. De la teoría general de grupos sabemos que todo grupo finito abeliano $\mathcal{G}$ puede descomponerse como la suma directa de grupos cíclicos

$$\mathcal{G} = \mathbb{Z}_{n_1} \times \mathbb{Z}_{n_2} \times \cdots \times \mathbb{Z}_{n_s},$$

donde $n_{i+1} \mid n_i$ para todo $i = 1, 2, \ldots, s-1$ y $n_s \geq 2$. Además ésta descomposición es única en el siguiente sentido: si

$$\mathcal{G} = \mathbb{Z}_{m_1} \times \mathbb{Z}_{m_2} \times \cdots \times \mathbb{Z}_{m_t},$$

es otra descomposición de $\mathcal{G}$ en la suma directa de grupos cíclico donde $m_{i+1} \mid m_i$ para todo $i = 1, 2, \ldots, t-1$ y $n_t \geq 2$ entonces $s = t$ y $n_i = m_i$ para toda $i = 1, 2, \ldots, s$. Decimos que $\mathcal{G}$ es un grupo abeliano de *tipo* $(n_1, n_2, \ldots, n_s)$ y *rango* s.

Teorema 4.4.6 *El grupo $E(\mathbb{F}_q)$ es un grupo abeliano de rango 1 ó 2. El tipo de un grupo es (n_1, n_2), es decir, $E(\mathbb{F}_q) \cong \mathbb{Z}_{n1} \times \mathbb{Z}_{n2}$, donde $n_2 \mid n_1$ y además $n_2 \mid q - 1$.*

Proposición 4.4.7 *Si el número de puntos $\#E(\mathbb{F}_q)$ es libre de cuadrados entonces $E(\mathbb{F}_q)$ es cíclico. También es cíclico si $(\#E(\mathbb{F}_q), q-1) = 1$ (ver [27]).*

En el caso de que el grupo formado por los puntos de una curva elíptica sea cíclico, se puede demostrar que dicha curva posee $\varphi(d_i)$ puntos generadores de orden d_i siendo d_i cada uno de los divisores del orden del número de puntos $\#E(\mathbb{F}_q)$. También, se puede demostrar que existe un único subgrupo de puntos para cada uno de los posibles órdenes. En particular, los puntos cuyo orden es igual al número de puntos de la curva se denominan *generadores de grupo*, puede que a partir de ellos y calculando sus sucesivos múltiplos se puede obtener la totalidad de los puntos de la curva.

Con todo lo anterior, es conveniente hacer notar que si el número de puntos de una curva elíptica es primo (y por lo tanto, libre de cuadrados) entonces el grupo aditivo formado por dichos puntos es cíclico y además cualquier punto de la curva (salvo el punto al infinito $\mathcal{O}$) es generador del mismo.

A continuación consideramos un ejemplo en donde el número de puntos es libre de cuadrados.

Ejemplo 4.4.8 *Sea la curva elíptica E dada por $y^2 = x^3+3$ sobre $\mathbb{F}_{29}$ y sea $\#E(\mathbb{F}_{29}) = 30$ (en la siguiente sección describimos como calcular $\#E(\mathbb{F}_q)$). Puesto que $\#E(\mathbb{F}_{29}) = 30 = 2 * 3 * 5$ es libre de cuadrados, el grupo formado por los puntos de la curva elíptica es cíclico, siendo los posibles órdenes de los mismo $\{1, 2, 3, 5, 6, 10, 15, 30\}$.*

4.4.2. Cálculo del número de puntos en una curva elíptica

Sea E la curva elíptica dada por la ecuación (4.5) y sea $\chi(x) = (\frac{x}{p})$ el carácter cuadrático de $\mathbb{F}_q$, donde $(\frac{x}{p})$ es el símbolo de Legendre. Deseamos usar χ para contar el número de puntos en una curva elíptica.

De esta manera, en todos los casos en número de puntos $y \in \mathbb{F}_q$ a la ecuación $y^2 = u$ es igual a $1 + \chi(u)$ y así el número de soluciones a (4.5), contando el punto al infinito es:

$$\#E(\mathbb{F}_q) = 1 + \sum_{x \in \mathbb{F}_q} (1 + \chi(f(x))) = 1 + q + \sum_{x \in \mathbb{F}_q} \chi(f(x)).$$

Utilizando el Teorema de Hasse 4.4.1 tenemos que

$$\#E(\mathbb{F}_q) = \mid \sum_{x \in \mathbb{F}_q} \chi(f(x)) \mid \leq 2\sqrt{q}.$$

Veamos un ejemplo práctico utilizando el campo $\mathbb{F}_{11}$.

Ejemplo 4.4.9 *Sea la curva elíptica E dada por $y^2 = x^3 + x + 6$ módulo $\mathbb{F}_{11}$. Entonces, para calcular los residuos cuadráticos utilizamos el Teorema 1.2.82. En la siguiente tabla reunimos los valores que toma $z = x^3 + x + 6$ para cada $x \in \mathbb{F}_{11}$ y el cálculo de sus respectivos residuos cuadráticos.*

x	0	1	2	3	4	5	6	7	8	9	10
$z = x^3 + x + 6$	6	8	5	3	8	4	8	4	9	7	4
$\left(\frac{z}{p}\right)$	-1	-1	$+1$	$+1$	-1	$+1$	-1	$+1$	$+1$	-1	$+1$

Por lo tanto tenemos que el número de puntos es:

$$\begin{aligned} \#E(\mathbb{F}_q) &= 11 + 1 + \sum_{x=0}^{10} \left(\frac{x^3 + x + 6}{11}\right) \\ &= 11 + 1 + 1 = 13. \end{aligned}$$

Para un cálculo exacto del número de puntos, el símbolo de Legendre debe evaluarse para toda $x \in \mathbb{F}_q$, lo cual es ineficiente para un campo finito grande $\mathbb{F}_q$. Como una alternativa a éste método está el algoritmo de Schoof [27, cap. 7], pero empieza a ser ineficiente cuando $q = p$ tiene más de 20 dígitos decimales. Sin embargo, en la actualidad se conocen otros algoritmos más eficientes para el cálculo del número de puntos de una curva elíptica sobre campos finitos de característica mayor que tres.

Para el caso de los campos de característica 2, Koblitz adaptó el algoritmo de Schoof a curvas elípticas sobre $\mathbb{F}_{2^n}$ (ver [19]).

4.4.3. Obtención de puntos sobre una curva elíptica

En esta sección describimos un método para encontrar puntos arbitrarios en una curva elíptica $E(\mathbb{F}_p)$ donde p es un primo impar y $p \equiv 3 \bmod 4$. Para el resto de los números primos impares puede consultarse [28, pag. 99].

El método más directo es elegir $x \in \mathbb{F}_p$ y ver si existe una y que satisfaga la ecuación (4.5); es decir, para cada x probamos si $z = f(x) = x^3 + ax + b$ es un residuo cuadrático. Supongamos que hemos encontrado una x para la cual estamos seguros de que existe tal y. Encontrar el valor de y implica resolver la congruencia $y^2 \equiv z \bmod p$. Si $p \equiv 3 \bmod 4$ el Teorema 1.2.82 implica que $y = z^{(p+1)/4}$, ya que $y^2 = z^{(p+1)/2} = zz^{(p-1)/2} \equiv z \bmod p$.

Tenemos un ejemplo práctico para encontrar los puntos de una curva elíptica.

Ejemplo 4.4.10 *Consideramos la curva $y^2 = x^3 + x + 6$ sobre el campo $\mathbb{F}_{11}$. Si variamos $x \in \mathbb{F}_{11}$, tenemos 11 posibilidades para y^2. Haciendo esto, encontramos 13 puntos (incluyendo el punto al infinito $\mathcal{O}$): $P_0 = \mathcal{O}, P_1 = (2,4), P_2 = (2,7), P_3 = (3,5), P_4 = (3,6), P_5 = (5,2), P_6 = (5,9), P_7 = (7,2), P_8 = (7,9), P_9 = (8,3), P_{10} = (8,8), P_{11} = (10,2), P_{12} = (10,9)$.*

En la siguiente tabla vemos que los puntos en la curva elíptica anterior forman un grupo abeliano respecto a la operación $\oplus$.

$\oplus$	P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	P_{11}	P_{12}
P_0	P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	P_{11}	P_{12}
P_1	P_1	P_6	P_0	P_7	P_{11}	P_2	P_{10}	P_8	P_4	P_5	P_{12}	P_9	P_3
P_2	P_2	P_0	P_5	P_{12}	P_8	P_9	P_1	P_3	P_7	P_{11}	P_6	P_4	P_{10}
P_3	P_3	P_7	P_{12}	P_9	P_0	P_{10}	P_8	P_5	P_2	P_6	P_4	P_1	P_{11}
P_4	P_4	P_{11}	P_8	P_0	P_{10}	P_7	P_9	P_1	P_6	P_3	P_5	P_{12}	P_2
P_5	P_5	P_2	P_9	P_{10}	P_7	P_{11}	P_0	P_{12}	P_3	P_4	P_1	P_8	P_6
P_6	P_6	P_{10}	P_1	P_8	P_9	P_0	P_{12}	P_4	P_{11}	P_2	P_3	P_5	P_7
P_7	P_7	P_8	P_3	P_5	P_1	P_{12}	P_4	P_2	P_0	P_{10}	P_{11}	P_6	P_9
P_8	P_8	P_4	P_7	P_2	P_6	P_3	P_{11}	P_0	P_1	P_{12}	P_9	P_{10}	P_5
P_9	P_9	P_5	P_{11}	P_6	P_3	P_4	P_2	P_{10}	P_{12}	P_8	P_0	P_7	P_1
P_{10}	P_{10}	P_{12}	P_6	P_4	P_5	P_1	P_3	P_{11}	P_9	P_0	P_7	P_2	P_8
P_{11}	P_{11}	P_9	P_4	P_1	P_{12}	P_8	P_5	P_6	P_{10}	P_7	P_2	P_3	P_0
P_{12}	P_{12}	P_{13}	P_{10}	P_{11}	P_2	P_6	P_7	P_9	P_5	P_1	P_8	P_0	P_4

En la figura 4 podemos visualizar el conjunto solución del ejemplo anterior como una nube de puntos. Esta es la razón por la que la suma geométrica de puntos no es aplicable a campos finitos.

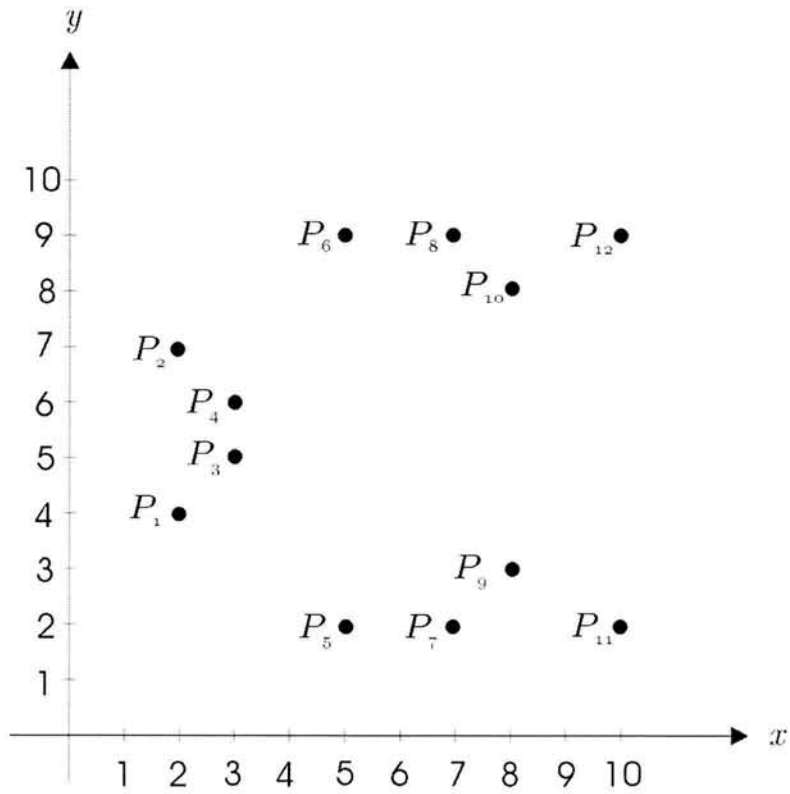

Figura 4: Conjunto solución.

Del ejemplo 4.4.10 se sigue que $\#E(\mathbb{F}_{11}) = 13$ y por lo tanto $E(\mathbb{F}_{11})$ es cíclico, entonces cualquier punto $P \in E(\mathbb{F}_{11})$ es un generador (excepto el punto al infinito $\mathcal{O}$). Por ejemplo, hemos calculado para $G = (3,5)$: $2G = (8,3) = P_9$, $3G = (5,9) = P_6$, $4G = (7,9) = P_8$, $5G = (2,7) = P_2$, $6G = (10,9) = P_{12}$, $7G = (10,2) = P_{11}$, $8G = (2,4) = P_1$, $9G = (7,2) = P_7$, $10G = (5,2) = P_5$, $11G = (8,8) = P_{10}$, $12G = (3,6) = P_4$, $13G = \mathcal{O}$.

Capítulo 5

Criptosistemas de curvas elípticas

Neal Koblitz [21] y Víctor Miller [30] propusieron en 1985 un criptosistema de clave pública, análogo al criptosistema ElGamal, en el cual el grupo $\mathbb{F}_q^*$ es reemplazado por el grupo que forman los puntos de una curva elíptica definida sobre un campo finito $\mathbb{F}_q$. Desde su introducción persiste una amplia discusión sobre su seguridad y eficiencia.

En el Capítulo 3 vimos como el grupo abeliano finito $\mathbb{F}_q^*$ (el grupo multiplicativo del campo finito $\mathbb{F}_q$) puede usarse para crear criptosistemas de clave pública. Más precisamente, fué la dificultad de resolver el problema del logaritmo discreto en campos finitos la que condujo a la discusión de los criptosistemas en el Capítulo 3. El propósito de este capítulo es hacer lo mismo, pero ahora los criptosistemas están basados en el grupo abeliano finito que forman los puntos de una curva elíptica E sobre un campo finito.

5.1. Múltiplos de puntos.

En curvas elípticas, multiplicar dos elementos en $\mathbb{F}_q^*$ es equivalente a *sumar* dos puntos en E, donde E es una curva elíptica sobre $\mathbb{F}_q$. De esta manera, elevar a la *n-ésima* potencia en $\mathbb{F}_q^*$ equivale a multiplicar el punto $P \in E$ por n, es decir, calcular el *n-ésimo* múltiplo de P. Como veremos más adelante, el cálculo de múltiplos de puntos es la operación principal de los criptosistemas de curvas elípticas y es por esta razón que se han realizado varios trabajos al respecto.

5.1.1. Método binario

Dada una curva elíptica E y un punto $P \in E$, deseamos calcular nP para algún entero no negativo n, es decir, queremos calcular

$$nP = P + P + \cdots + P.$$

Teorema 5.1.1 *Sea $(b_{N-1} \ldots b_2 b_1 b_0)_2$ la representación binaria de n. Definimos la sucesión de puntos $P_1, \ldots, P_N$ como sigue,*

$$P_1 = P$$

$$P_i = \begin{cases} 2P_{i-1} & si\ b_{N-i} = 0 \\ 2P_{i-1} + P & si\ b_{N-i} = 1 \end{cases} \tag{5.1}$$

para $i = 2, \ldots, N$. *Entonces* $P_N = nP$.

De la ecuación (5.1) deducimos que P_i se obtiene a partir de la *duplicación* y *suma* de puntos en E.

Ejemplo 5.1.2 *Para encontrar* $100P$, *escribimos* $100 = (1100100)_2$ *y utilizando la fórmula (5.1) tenemos:*

$P_1 = P,\ P_2 = 2P + P$	$P_2 = \begin{cases} 2P_1 & si\ b_5 = 0 \\ 2P_1 + P & si\ b_5 = 1 \end{cases}$
$P_3 = 2(2P + P)$	$P_3 = \begin{cases} 2P_2 & si\ b_4 = 0 \\ 2P_2 + P & si\ b_4 = 1 \end{cases}$
$P_4 = 2(2(2P + P))$	$P_4 = \begin{cases} 2P_3 & si\ b_3 = 0 \\ 2P_3 + P & si\ b_3 = 1 \end{cases}$
$P_5 = 2(2(2(2P + P))) + P$	$P_5 = \begin{cases} 2P_4 & si\ b_2 = 0 \\ 2P_4 + P & si\ b_2 = 1 \end{cases}$
$P_6 = 2(2(2(2(2P + P))) + P)$	$P_6 = \begin{cases} 2P_5 & si\ b_1 = 0 \\ 2P_5 + P & si\ b_1 = 1 \end{cases}$
$P_7 = 2(2(2(2(2(2P + P))) + P))$	$P_7 = \begin{cases} 2P_6 & si\ b_0 = 0 \\ 2P_6 + P & si\ b_0 = 1 \end{cases}$

Por lo tanto $100P = 2(2(2(2(2(2P + P))) + P))$.

En particular, si E es una curva de Koblitz se utiliza el método de *expansión binaria balanceada* propuesto por Koblitz en [19].

5.2. Asignación de mensajes a puntos de una curva elíptica

Ahora nuestro objetivo es codificar las unidades texto claro con puntos en alguna curva elíptica $E(\mathbb{F}_q)$. Con esta codificación no estamos cifrando el mensaje, sólo estamos etiquetando el texto en claro con elementos de una curva elíptica.

Pero antes debemos hacer dos observaciones: En primer lugar, no se conoce un algoritmo determinístico de tiempo polinomial para listar un gran número de puntos en una curva elíptica arbitraria $E(\mathbb{F}_q)$. Sin embargo, hay algoritmos probabilísticos en donde la oportunidad de fracasar es pequeña, como veremos a continuación. En segundo lugar, no es suficiente generar puntos aleatorios en E: de manera que para codificar un gran número de posibles unidades de mensaje m, necesitamos una forma sistemática para generar puntos que estén relacionados de alguna manera con m, por ejemplo, que la coordenada x tenga una relación simple con el entero m.

Enseguida describimos un método probabilístico para etiquetar nuestras unidades de mensaje m con puntos de una curva elíptica $E(\mathbb{F}_q)$, ver [17, pag. 179].

Sea k un entero tal que $30 \leq k \leq 50$, como lo sugiere la práctica. Supongamos que nuestras unidades de mensaje m están en el intervalo $0 \leq m < M$ y que elegimos un campo finito $\mathbb{F}_q$ tal que $q > Mk$. Entonces existe una función inyectiva de $\{1, \ldots, Mk\}$ a $\mathbb{F}_q$. Escribimos a los enteros $1, \ldots, Mk$ como $l = mk + j$, donde $1 \leq j \leq k$.

Para cada m, al variar j obtenemos k números l, hacemos $x_m = l$ y calculamos el lado derecho de la ecuación

$$y^2 = f(x) = x_m^3 + ax_m + b.$$

Si $f(x)$ tiene raíz cuadrada (ver Sección 4.4.3) entonces nos detenemos, de lo contrario avanzamos a $j + 1$. Si encontramos una y_m tal que $y_m^2 = f(x)$, etiquetamos a la unidad de mensaje m con el punto $P_m = (x_m, y_m)$.

Siempre que encontremos una x_m para el cual $f(x)$ es un cuadrado antes de que j sea más grande que k, podemos recuperar m (texto claro) del punto (x_m, y_m) con la fórmula $m = [(x_m - 1)/k]$, donde x_m es el entero correspondiente a x bajo la correspondencia inyectiva entre los enteros l y los elementos de $\mathbb{F}_q$.

El éxito del algoritmo anterior se debe a la siguiente proposición.

Proposición 5.2.1 *Dado un entero $m > 0$, la probabilidad de no encontrar el punto P_m en la curva elíptica es $1/2^k$.*

Ejemplo 5.2.2 *Sea $k = 30$, $M = 30$ y $\mathbb{F}_q$ con $q > 900$, a saber $\mathbb{F}_{907}$, así:*

$m = 0$	1	2	...	30	j
$m = 1$	31	32	...	60	$k + j$
$m = 2$	61	62	...	90	$2k + j$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$m = M - 1$	$(M-1)k + 1$	$(M-1)k + 2$	...	kM	$(M-1)k + j$

Consideramos el alfabeto $\mathcal{A} = \{A, B, C, \ldots, Z\}$ y la curva elíptica E dada por la ecuación $y^2 = x^3 + x + 1 \in \mathbb{F}_{907}[x]$. En este caso $f(x) = x^3 + x + 1$, y el algoritmo queda de la siguiente manera:

1. *Para identificar al carácter A con un punto de la curva, tomamos $m = 0$, $j = 1$ entonces $x_A = l = 1$. Como $f(1) = 3$ y éste no tiene raíz cuadrada en $\mathbb{F}_{907}$ tomamos otra j; a saber, sea $j = 9$ entonces $x_A = l = 9$ y como $f(9) = 739$ tiene por raíz cuadrada a 620, tenemos que $P_A = (9, 620)$.*

2. *Se continúa de la misma forma para el resto de los caracteres del alfabeto $\mathcal{A}$.*

5.3. El problema del logaritmo discreto en curvas elípticas

En la sección 3.3 discutimos criptosistemas de clave pública basados en el problema del logaritmo discreto en el grupo multiplicativo de $\mathbb{F}_q$. Ahora damos una definición análoga para el grupo (bajo la suma de puntos) que forman los puntos de una curva elíptica $E(\mathbb{F}_q)$.

Definición 5.3.1 *Dada una curva elíptica $E(\mathbb{F}_q)$, un punto $B \in E$ de orden n y un punto $P \in E$. El problema del logaritmo discreto en E (respecto a la base B) es encontrar un entero x, $0 \leq x \leq n-1$, tal que $P = xB$ si tal entero existe.*

En general, el cálculo del entero x no es un problema fácil, sobre todo cuando se trabaja con campos de dimensión elevada.

La similitud entre las definiciones del problema del logaritmo discreto y el problema del logaritmo discreto en curvas elípticas, hacen que todos los criptosistemas basados en el problema del logaritmo discreto puedan adaptarse utilizando curvas elípticas. Así, tenemos variantes de los criptosistemas anteriores convertidos a curvas elípticas: Diffie-Hellman-CE, ECDSA, ElGamal-CE, Massey-Omura EC. Para adaptarlos al grupo de curvas elípticas se hacen algunas modificaciones, pero los principios fundamentales son los mismos.

Es probable que el problema del logaritmo discreto en curvas elípticas sea más difícil que el problema del logaritmo discreto en campos finitos. Las técnicas más fuertes desarrolladas para campo finitos parecen no funcionar en curvas elípticas.

Hasta 1990 los únicos algoritmos de logaritmo discreto conocidos para curvas elípticas eran aquellos que funcionan para cualquier grupo, sin tomar en consideración ninguna estructura, los cuales son algoritmos de tiempo exponencial (éstos algoritmos se basan en la factorización del orden del grupo). Pero en 1993 **M**enezes, **O**kamoto y **V**anstone (conocido como **MOV** [29]) encontraron una nueva aproximación para el problema del logaritmo discreto en curvas elípticas en $E(\mathbb{F}_q)$, en el cual utilizaron la paridad de Weil [40, pag. 132] para "encajar" el grupo E en el grupo multiplicativo de alguna extensión del campo $\mathbb{F}_{q^k}$. De esta manera, al utilizar este morfismo se reduce el problema del logaritmo discreto sobre E al problema del logaritmo discreto en $\mathbb{F}^*_{q^k}$.

Sin embargo, para que funcione la reducción de la paridad de Weil es indispensable que $k \leq 6$. Las únicas curvas elípticas para las cuales k es pequeño esencialmente son las curvas elípticas *supersingulares*, ejemplos de este tipo de curvas son las de la forma

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

con car $\mathbb{F}_q = p \equiv -1 \bmod 4$ y las curvas de la forma

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

con car $\mathbb{F}_q = p \equiv -1 \bmod 3$.

Sin embargo, la mayoría de la curvas elípticas son no-supersingulares y no se aplica el algoritmo de tiempo subexponencial propuesto por **MOV**.

Entonces, si se desea evitar el ataque **MOV** para este tipo de curvas es importante trabajar en un campo suficientemente grande, donde el trabajo computacional para calcular el problema del logaritmo discreto sea lento. Sin embargo, esto implica que al intentar ganar seguridad en los criptosistemas se sacrifica eficiencia, ya que el trabajo computacional no sólo se incrementa para el criptoanalista, sino también para la entidad que cifra y descifra.

Mencionamos anteriormente que otro tipo de ataque para resolver el problema del logaritmo discreto en curvas elípticas se basa en la factorización de $N = \#(E(\mathbb{F}_q))$. De esta manera, si se elige una curva no-supersingular donde N tenga un factor primo lo suficientemente grande, se podrá tener una seguridad aceptable. Actualmente, suponiendo un ataque con los mejores algoritmos conocidos trabajando en paralelo, se estima que $N = \#(\mathbb{F}_q)$ será seguro siempre que tenga un factor primo de por lo menos 50 dígitos decimales (es decir, para sea difícil resolver el problema del logaritmo discreto es importante elegir apropiadamente una curva elíptica E y un campo $\mathbb{F}_q$ tal que $\#E(\mathbb{F}_q)$ sea divisible por un primo lo suficientemente grande ó que q sea un primo grande de aproximadamentes 160 bits, ver [23]) .

Hasta ahora hemos visto que a diferencia del problema del logaritmo discreto y de la factorización de enteros, no se conocen algoritmos de tiempo subexponencial para el problema del logaritmo discreto en curvas elípticas, salvo el caso del algoritmo **MOV**.

De esta manera para que resulte difícil resolver el problema del logaritmo discreto es importante elegir apropiadamente una curva elíptica E y un campo q tal que $\#E(\mathbb{F}_q)$ sea divisible por un primo lo suficientemente grande ó que q sea un primo grande.

5.4. El análogo de Diffie-Hellman

Supongamos que **A**le y **B**ob desean intercambiar una clave secreta a través de un canal inseguro para posteriormente usarla en criptosistemas convencionales. Primero eligen un campo finito $\mathbb{F}_q$ y una curva elíptica $E(\mathbb{F}_q)$, los cuales son de conocimiento público.

Ale y **B**ob primero eligen un punto $B \in E$ como la "base". El punto B cumple el mismo papel que el generador g en el sistema de Diffie-Hellman en campos finitos. Sin embargo, no necesitamos insistir que B sea un generador del grupo de puntos en una curva elíptica E, dado que el grupo puede no ser cíclico. Incluso si es cíclico, queremos evitar el esfuerzo de verificar que B sea un generador (ó incluso determinar el número de puntos en $(\mathbb{F}_q)$). Nos gustaría que la cardinalidad del subgrupo generado por B sea

grande, de preferencia que sea casi del mismo orden de magnitud que $\#E(\mathbb{F}_q)$. Por ahora, supongamos que $B \in E$ es un punto fijo conocido por todos los usuarios del sistema, cuyo orden es muy grande (N ó un divisor grande de N).

Para generar una clave, **A**le primero elige un entero aleatorio a del orden de magnitud q, que mantiene en secreto. Luego calcula

$$P_A = aB \in E,$$

(se suma B consigo mismo a veces, lo cual arroja otro punto P_A sobre la curva) y lo publica en un directorio.

Bob hace lo mismo: elige un entero aleatorio b y publica

$$P_B = bB \in E.$$

Ale calcula el punto secreto $P_{AB} = aP_B = abB$, mientras que **B**ob puede calcular $P_{BA} = bP_A = baB$. Puesto que el grupo de puntos de una curva elíptica es abeliano se verifica que $P_{AB} = P_{BA}$, de esta manera pueden utilizar éste punto como su clave secreta para utilizarla en posteriores comunicaciones cifradas utilizando criptosistemas convencionales.

Ambos usuarios pueden calcular ésta clave secreta. Por ejemplo, **A**le conoce bB (de conocimiento público) y su propia clave secreta a. Sin embargo una tercera persona sólo conoce aB y bB. Al parecer no hay una forma para calcular abB conociendo sólo aB y bB sin resolver el problema del logaritmo discreto en curvas elípticas (determinar a conociendo B y aB).

5.5. El análogo del Massey-Omura

Como en el caso de campos finitos, éste es un criptosistema de clave pública para transmitir unidades de mensajes m, que están etiquetadas con puntos P_m en alguna curva elíptica $E(\mathbb{F}_q)$ fija y públicamente conocida.

Supongamos que hemos calculado $N = \#E(\mathbb{F}_q)$ (y que es de conocimiento público). Cada usuario del sistema selecciona de manera secreta un entero e entre 1 y N tal que $(e, N) = 1$ y calcula $d = e^{-1} \bmod N$.

Si **A**le quiere enviarle el mensaje P_m a **B**ob, primero le envía el punto

$$e_A P_m \in E.$$

Esto no le dice nada a **B**ob porque no puede recuperar P_m sin el conocimiento de d_A ni e_A. Pero sin intentar entenderlo, lo multiplica por su clave privada e_B y le envía a **A**le el punto

$$e_B e_A P_m \in E.$$

El tercer paso consiste en que **A**le multiplique el punto $e_B e_A P_m$ por d_A. Ya que $NP_m = \mathcal{O}$ y $d_A e_A \equiv 1 \bmod N$, esto da como resultado el punto $e_B P_m$, el cual **A**le le regresa a **B**ob, quien ahora puede leer el mensaje al multiplicar el punto $e_B P_m$ por d_B.

Notamos que un intruso puede llegar a conocer $e_A P_m$, $e_B e_A P_m$ y $e_B P_m$. Si el intruso pudiera resolver el problema del logaritmo discreto en curvas elípticas entonces podría determinar e_B a partir de los primeros dos puntos y calcular $d_B = e_B^{-1} \bmod N$ y $P_m = d_B (e_B P_m)$.

5.6. El análogo al criptosistema ElGamal

Éste es otro criptosistema de clave pública para transmitir unidades de mensaje P_m, es equivalente al criptosistema ElGamal descrito en la Sección 3.9. Al igual que en los sistemas de distribución de claves descrito con anterioridad, empezamos con un campo finito $\mathbb{F}_q$ públicamente conocido, una curva elíptica $E(\mathbb{F}_q)$ y el punto base $B \in E$. Cada usuario elige un entero aleatorio a, el cual se mantiene en secreto, calcula y publica el punto aB.

Para enviarle el mensaje P_m a **B**ob, **A**le elige un entero aleatorio k y envía el par de puntos

$$(kB, P_m + k(a_B B)),$$

donde $a_B B$ es la clave pública de **B**ob. Para leer el mensaje, **B**ob multiplica kB por su clave privada a_B y resta el resultado al segundo punto:

$$P_m + k(a_B B) - a_B(kB) = P_m.$$

De esta manera, **A**le envía un disfraz a P_m junto con una pista kB, el cual es suficiente para quitar la máscara $k(a_B B)$ si uno conoce la clave secreta a_B. Un intruso que resuelva el problema del logaritmo discreto en curvas elípticas, puede determinar a_B de la información públicamente conocida B y $a_B B$.

5.7. El algoritmo de firma digital con curvas elípticas

El **ECDSA** (**E**lliptic **C**urve **C**ryptographic **D**igital **S**ignature) es el análogo al **DSS**, descrito en la Sección 3.10. En 1999 el **ECDSA** [14] fué adoptado como un estándar ANSI y en el año 2000 hicieron lo mismo las organizaciones IEEE y la NIST.

Como antes, elegimos un campo finito $\mathbb{F}_q$ que es de conocimiento público, una curva elíptica $E(\mathbb{F}_q)$ y un punto base $P \in E$ de orden n con $n > 2^{160}$ y $n > 4\sqrt{q}$. Un usuario **A** primero elige un número entero aleatorio d en el intervalo $[1, n-1]$ y calcula $Q = dP$. La clave pública de **A** es el punto Q, mientras que su clave secreta es el entero d.

Ahora, si **A**le quiere firmar una unidad de mensaje m hace lo siguiente. Primero aplica la función hash SHA-1 (ver [11]) a su mensaje, obteniendo el correspondiente valor hash h. Luego elige un entero aleatorio k en el intervalo $[1, n-1]$, calcula

$$kP = (x_1, y_1)$$

y hace

$$r = x_1 \bmod n.$$

Si resulta que $r = 0$ entonces debe elegir otro entero k. Finalmente **A**le calcula k^{-1} mod n para realizar el siguiente cálculo:

$$s \equiv k^{-1}(h + dr) \bmod n,$$

si $s = 0$ entonces debe buscar otra k. Después de realizar los cálculos anteriores, la firma de **A** para el mensaje m es el par (r, s).

Si un usuario **B** desea verificar la firma (r, s) de **A**le sobre m, debe hacer lo siguiente: primero debe obtener una copia auténtica de Q (la clave pública de **A**), verificar que los enteros r, s estén en el intervalo $[1, n-1]$. Posteriormente calcula el valor hash h del mensaje y calcula

$$\begin{aligned} u_1 &= s^{-1}h \bmod n, \\ u_2 &= s^{-1}r \bmod n. \end{aligned}$$

Luego, realiza el siguiente cálculo

$$X = u_1 P + u_2 Q.$$

Si $X = \mathcal{O}$ entonces rechaza la firma. De lo contario, calcula

$$v = x_1 \bmod n$$

donde $X = (x_1, y_1)$. **B**ob acepta la firma si y sólo si $v = r$.

A continuación vemos que la prueba de la firma funciona. Si la firma (r, s) sobre el mensaje m fué generado por **A**, entonces $s = k^{-1}(h+dr) \bmod n$. Reorganizando tenemos:

$$\begin{aligned} k &\equiv s^{-1}(h + dr) \equiv s^{-1}h + s^{-1}dr, \\ k &\equiv u_1 + u_2 d \bmod n. \end{aligned}$$

De esta manera $u_1 P + u_2 Q = u_1 P + u_2 dP = (u_1 + u_2 d)\, P = kP$, así $v = r$ como se requirió.

La única diferencia entre el **DSS** y el **ECDSA** es la generación de r. En la sección 3.10 se obtiene al elegir aleatoriamente un entero k y luego se reduce módulo q, de esta manera obtuvimos un entero en el intervalo $[1, q-1]$. Aquí, hemos visto que el entero r en el intervalo $[1, n-1]$ se genera al elegir la coordenada x de un punto aleatorio kP y reducirlo módulo n.

5.8. Aspectos a considerar en los CCE

En esta sección describimos algunos de los puntos que se deben tener en cuenta para implementar criptosistemas basados en el problema del logaritmo discreto en curvas elípticas.

5.8.1. Elección del campo finito

Un aspecto importante en criptografía de curvas elípticas es seleccionar un campo base en donde se puedan efectuar operaciones aritméticas de manera eficiente. Hay dos tipos de campos finitos $\mathbb{F}_q$ que son apropiados para los criptosistemas de clave pública, en particular para los criptosistemas de curvas elípticas: los campos primos $\mathbb{F}_p$ y los campos binarios $\mathbb{F}_{2^m}$.

El campo finito $\mathbb{F}_p$

Sabemos que el campo $\mathbb{F}_p$ contiene p elementos y que para para cada primo impar p hay sólo un campo primo $\mathbb{F}_p$. También que los elementos de $\mathbb{F}_p$ se representan por el conjunto de enteros $\{0, 1, \ldots, p-1\}$ (por la definición 1.2.31), los cuales se escriben en expansión binaria.

Para facilitar la operabilidad, el campo primo $\mathbb{F}_p$ debe estar sujeto a las siguientes restricciones:

$$\lceil \log_2 p \rceil \in \{112, 128, 160, 192, 224, 256, 384, 521\}.$$

Los campos finitos primos $\mathbb{F}_p$ recomendados por la NIST en [33] son: $p = 2^{192} - 2^{64} - 1$, $p = 2^{224} - 2^{96} + 1$, $p = 2^{256} - 2^{224} + 2^{192} + 2^{96} - 1$, $p = 2^{384} - 2^{128} - 2^{96} + 2^{32} - 1$ y $p = 2^{521} - 1$.

En ese artículo se recomiendan las curvas elípticas aleatorias $y^2 = x^3 - 3x + b$ para cada uno de estos campos primos y se denotan por **P-192**, **P-224**, **P-256**, **P-384** y **p-521**, respectivamente (también se pueden consultar en [14]).

En [3] se presenta una implementación de un software utilizando las curvas elípticas recomendadas por la NIST sobre campos primos.

El campo finito $\mathbb{F}_{2^m}$

Los campos finitos de la forma $\mathbb{F}_{2^m}$ con $m \geq 1$ son convenientes de usar, puesto que sus elementos encajan a la perfección en una palabra de datos de longitud de m bits. Esto es porque los elementos de $\mathbb{F}_{2^m}$ son representados por el conjunto de polinomios binarios de grado menor o igual que $m-1$(representación polinomial, sección 1.2.2):

$$\mathbb{F}_{2^m} = \left\{ a_{m-1}x^{m-1} + a_{m-2}x^{m-2} + \cdots + a_1 x + a_0 : a_i \in \{0, 1\} \right\}$$

donde estos polinomios se representan en la computadora por la palabra de datos de m bits $(a_{m-1}a_{m-2}\cdots a_1 a_0)$.

Los campos $\mathbb{F}_{2^m}$ están sujetos a las siguientes restricciones:

$$m \in \{113, 131, 163, 193, 233, 239, 283, 409, 571\}.$$

La suma y la resta en el campo $\mathbb{F}_{2^m}$ se realizan utilizando un polinomio irreducible. Los polinomios irreducibles recomendados (ver [39]) se muestran en la siguiente tabla.

Campo	Polinomio irreducible
$\mathbb{F}_{2^{113}}$	$f(x) = x^{113} + x^9 + 1$
$\mathbb{F}_{2^{131}}$	$f(x) = x^{131} + x^8 + x^3 + x^2 + 1$
$\mathbb{F}_{2^{163}}$	$f(x) = x^{163} + x^7 + x^6 + x^3 + 1$
$\mathbb{F}_{2^{193}}$	$f(x) = x^{193} + x^{15} + 1$
$\mathbb{F}_{2^{233}}$	$f(x) = x^{233} + x^{74} + 1$
$\mathbb{F}_{2^{239}}$	$f(x) = x^{239} + x^{36} + 1$ ó $x^{239} + x^{158} + 1$
$\mathbb{F}_{2^{283}}$	$f(x) = x^{283} + x^{12} + x^7 + x^5 + 1$
$\mathbb{F}_{2^{409}}$	$f(x) = x^{409} + x^{87} + 1$
$\mathbb{F}_{2^{571}}$	$f(x) = x^{571} + x^{10} + x^5 + x^2 + 1$

5.9. Conclusiones

En esta sección tratamos de resumir los resultados mas importantes acerca de la criptografía de clave pública con curvas elípticas.

Hasta ahora el problema del logaritmo discreto en curvas elípticas parece ser más difícil que el problema de factorización de enteros grandes y el problema del logaritmo discreto, dado que para el primero no se ha encontrado un algoritmo de tiempo subexponencial que lo resuelva (salvo para las curvas elípticas supersingulares).

Es posible utilizar una curva elíptica cuyo grupo de puntos sea significativamente más pequeño que el utilizado en $\mathbb{F}_p^*$ para el problema del logaritmo discreto (o el problema de factorizar un número entero compuesto).

Los criptosistemas de curvas elípticas son una alternativa a los otros criptosistemas descritos en este trabajo, en lugar de reemplazarlos. Los CCE tienen grandes ventajas, especialmente cuando se utilizan en dispositivos en donde la capacidad de almacenamiento y procesamiento es restringida. Las aplicaciones típicas son: smart cards, e-commerce, aplicaciones bancarias.

Un tipo especial de curvas no-supersingulares son las curvas de Koblitz, en la actualidad éste tipo de curvas son las más utilizadas para implementar criptosistemas elípticos, debido a que presentan buenas propiedades criptográficas:

1. Son no-supersingulares y por lo tanto no puede aplicarse eficientemente el algoritmo **MOV**.

2. El número de puntos sobre la curva tiene un factor primo grande (no pueden calcularse el logaritmo discreto de manera eficiente con los mejores algoritmos).

3. Existe un gran variedad de curvas elípticas de este tipo, por lo que son fáciles de hallar.

4. Es eficiente calcular múltiplos de puntos (sumar un punto consigo mismo).

Bibliografía

[1] Adleman L., Rivest R., Shamir A., *A Method for Obtaining Digital Siganatures and Public-key Cryptosystem*, Communications of the ACM, 21, 120-126, 1978.

[2] Brieskorn E., Knörrer H., *Plane Algebraic Curves*, Birkhäuser Verlag, 1986.

[3] Brown, M., Hankerson D., López J., Menezes A., *Software implementation of the NIST Elliptic Curves over Prime Field*, LNCS 2020, pp. 250-265, Springer Verlag, 2001.

[4] Certicom. *Elliptic Curve Cryptography*, September 2000. http://www.certicom.com.

[5] Childs Lindsay N., *A Concrete Introduction to Higher Algebra*, Second Edition, Springer Verlag, 1995.

[6] Cohen Henry, *A Course in Computational Algebraic Number Theory*, Springer Verlag, U.S.A. 1996.

[7] Coppersmith D., *Fast Evaluation of Logarithms in fields of characteristic two*, IEEE Trans. Inform. Theory, IT-30 (4), 587-594, July 1984.

[8] Diffie W., Hellman M., *New Directions in Cryptography*, IEEE Transactions on Information Theory, IT- 22, 644-654, 1976.

[9] Dobbertin, H., Bosselaers A., Preneel B., *RIPEMD-160: A Strengthened Version of RIPEMD*. http://www.esat.kuleuven.ac.be/~cosicart/pdf/AB-9601/.

[10] ElGamal T., *A Public Key Cryptosistem and a Signature Scheme Bases on Discrete Logarithms*, Advances in Cryptology-CRYPTO'84, Lectures Notes in Computer Science 196, Springer-Verlag, 10-18, 1985.

[11] Federal Information Processing Standards (FIPS) Publication 180-2, *Secure Hash Standard (SHS)*, DoC/NIST, U.S.A. 2002.

[12] Fraleigh J.B., *Algebra abstracta*, Addison-Wesley Iberoamericana. 1987.

[13] Husemöller D., *Elliptic Curves*, Springer Verlag, New York,1987.

[14] Johnson D., Menezes A., Vastone S., *The Elliptic Curve Digital Signature (ECDSA)*, Technical Report CORR 99-34, Dept. of C&O, University of Waterloo, Canada 2000.

[15] Joux A., Lercier R., *Discrete logarithms in GF(p) (120 decimal digits)* http://listserv.nodak.edu/archivies/nmbrthry.html, Abril 2001.

[16] Kahn D., *The Codebreakers, the Story of Secret Writing*, Macmillan, 1967.

[17] Koblitz N., *A Course in Number Theory and Cryptography*, Springer Verlag, New York, 1994.

[18] Koblitz N., *Algebraic Aspects of Cryptography*, Springer Verlag, Berlin, 1998.

[19] Koblitz N., *CM-Curves with good cryptographic properties*, Advances of Cryptology, CRYPTO'91, LNCS 1294, 357-371, 1997.

[20] Koblitz, N., *Constructing Elliptic Curve Cryptosystems in Characteristic 2*, Advances in Cryptology-CRYPTO'90, Lectures Notes in Computer Science, 537, Springer-Verlag, 156-167, 1991.

[21] Koblitz N., *Elliptic Curve Cryptosystems*, Mathematics of Computation, 48, 203-209, 1987.

[22] Koblitz N., *Introduction to Elliptic Curves and Modular Forms*, Springer Verlag, 1993.

[23] Koblitz, N., Menezes, A., Vanstone, S., *The State of Elliptic Curve Cryptographic*, Designs, Codes and Cryptography, Vol 19, 173-193, Kluwer Academic Publishers, Boston, 2000.

[24] Koshy T., *Elementary Number Theory with Applications*, Academic Press, U.S.A, 2002.

[25] Lidl R., Niederreiter H., *Introduction to Finite Fields and their Applications*, Cambridge University Press, U.S.A. 2000.

[26] McEliece R.J., *Finite Fields for Computer Scientist and Engineers*, Kluwer Academic Publishers, U.S.A 1987.

[27] Menezes A.J., *Elliptic Curve Public Key Cryptosystems*, Kluwer Academic Publishers, 1993.

[28] Menezez A. J., *Handbook of Applied Cryptography.* CRC 2000.

[29] Menezes A., Okamoto T., Vanstone., *Reducing Elliptic Curves Logarithms in a finite field*, IEEE Transactions on Information Theory, 39, 1639-1646, 1993.

[30] Miller V., *Uses of Elliptic Curves in Cryptography*, Advances in Cryptology: Proceedings of Crypto '85, Lectures Notes in Computer Science, 218, 417-426, Springer Verlag, 1986.

[31] Mollin R. A., *An Introduction to Cryptography*, Chapman & Hall/CRC, U.S.A 2001.

[32] Mollin R. A., *RSA and Public-Key Cryptography*, Chapman & Hall/CRC, U.S.A 2003.

[33] National Institute of Standars and Technology, *Recommended Elliptic Curves for Federal Government Use*, May 1999; revised July 1999. http://csrc.nist.gov/encryption.

[34] Pastor F. M., *Criptografía digital*, Prensas Universitarias de Zaragoza, Zaragoza, 1998.

[35] Preneel B., *The State of Cryptographic Hash Functions*, Lectures on Data Security., Lectures Notes in Computer Science., pag 158-170. Springer Verlag, 1999.

[36] Rivest R., *The MD5 message digest algorithm*, Request for Comments (RFC) 1810, Internet Activities Board, Internet Privacy Task Force, June 1995.

[37] Rukhin A., Soto J., et al, *A Statistical Test Suite for Random and Psedorandom Number Generators for Cryptographic Applications*, NIST Special Publication 800-22, 2001.

[38] Saloma A., *Public Key Cryptography*, Springer Verlag, Berlin 1996.

[39] SEC 1, *Elliptic Curve Cryptography*, Standars for Efficient Cryptography Group, September, 2000. http://www.secg.org/

[40] Silverman J. H., *The Arithmetic of Elliptic Curves*, Springer Verlag, New York, 1992.

[41] Silverman J. H., *Rational Points on Elliptic Curves*, Springer Verlag, New York, 1992.

[42] Stallings W., *Network and Internetwork Security Principles and Practice*, Prentice Hall, 1994.

[43] Stinson D. R., *Cryptography: Theory and Practice*, CRC Press LLC, U.S.A 1995.

[44] Thomé, E., *Computation of Discrete Logarithms in* $\mathbb{F}_{2^{607}}$, Laboratoire d'Informatique (LIX). http://www.loria.fr/~thome/announcement/announcement.html. Sep. 2002.

[45] Waterhouse, E. *Abelian varieties over finite fields*, Ann. Sci. École Norm. Sup., 2, 521-560, 1969.