Universidad de Valladolid

FACULTAD DE CIENCIAS

TRABAJO FIN DE GRADO

Grado en Matemáticas

EL PROBLEMA DEL LOGARITMO DISCRETO EN CURVAS ELÍPTICAS

Autora: Sara Cabrero del Campo
Tutor: Umberto Martínez Peñas
Año: 2024/2025

Resumen:

Los criptosistemas basados en el problema del logaritmo discreto, como por ejemplo el criptosistema de ElGamal, son ampliamente utilizados hoy en día en la práctica. Sin embargo, la propuesta original, basada en el grupo multiplicativo (cíclico) de un cuerpo finito, es vulnerable a ciertos ataques, como el Index Calculus. Como alternativa, Koblitz y Miller propusieron el grupo asociado a una curva elíptica sobre un cuerpo finito. Dicho grupo ha sido capaz de resistir ataques como el Index Calculus, al mismo tiempo que resulta eficiente de implementar en la prática.
En este trabajo, se estudiará la estructura del grupo de una curva elíptica sobre un cuerpo finito y su implementación para criptosistemas basados en el problema del logaritmo discreto.

Palabras clave: Curva elíptica, logaritmo discreto, criptosistemas, clave pública

Abstract:

Cryptosystems based on the discrete logarithm problem, such as the ElGamal cryptosystem, are widely used in practice today. However, the original proposal, based on the multiplicative (cyclic) group of a finite field, is vulnerable to certain attacks, such as Index Calculus. As an alternative, Koblitz and Miller proposed the group associated to an elliptic curve on a finite field. Such a group has been able to resist attacks such as the Index Calculus, while being efficient to implement in practice. In this paper, we will study the structure of the group of an elliptic curve over a finite field and its implementation for cryptosystems based on the discrete logarithm problem.

Keywords: Elliptic curve, discrete logarithm, cryptosystems, public key

Índice general

Introducción

El objetivo de este trabajo es presentar los principales criptosistemas basados en el logaritmo discreto,con un enfoque particular en aquellos que operan en grupos asociados a curvas elípticas sobre cuerpos finitos. Los criptosistemas basados en el logaritmo discreto en grupos generales son vulnerables a ciertos ataques, como el Index Calculus. Por ello, se propone, en su lugar, el uso de grupos definidos sobre curvas elípticas sobre cuerpos finitos, como introdujeron Koblitz y Miller. Esta alternativa ofrece ventajas tanto en términos de seguridad como de eficiencia en los criptosistemas.

En el trabajo usamos herramientas de la teoría de números y la geometría diferencial, para definir las curvas elípticas; teoría de grupos, para estudiar las propiedades del grupo de curvas elípticas sobre cuerpos finitos; y métodos criptográficos, para analizar los criptosistemas y sus posibles ataques.

El trabajo se organiza en tres capítulos. En el primero, dedicado a los preliminares, comenzaremos desarrollando el espacio proyectivo y las curvas algebraicas, que nos servirá de base para poder afrontar la teoría de curvas elípticas.

En el segundo capítulo abordaremos la matemática de las curvas elípticas, que sirve como base a cualquier aplicación práctica que se verá posteriormente. En él, se definen y demuestran, desde un punto de vista aritmético y geométrico, las propiedades fundamentales de las curvas elípticas. Se desarrolla la forma de Weierstrass de una curva elíptica. El resultado clave de este capítulo es la definición de la estructura de grupo asociada a estas curvas. Finalmente, se aborda cómo trabajar con este grupo de manera que se puedan aplicar los conocimientos adquiridos al campo de la criptografía. Nos centraremos en el caso de curvas definidas sobre cuerpos de característica distinta de dos y tres, ya que, aunque no existen muchas diferencias conceptuales, este enfoque nos permite utilizar la forma de Weierstrass. En esta sección es importante destacar el teorema de Hasse, que nos da una aproximación al número de puntos de una curva elíptica.

El último capítulo aplica todo lo aprendido en los capítulos anteriores para definir diversos métodos criptográficos, haciendo uso práctico de las curvas elípticas. El problema central de este capítulo es el estudio de los métodos criptográficos basados en el problema del logaritmo discreto en el grupo de curvas elípticas.

En el capítulo se explican los algoritmos para grupos cíclicos en general y después para subgrupos cíclicos del grupo de una curva elíptica. En este capítulo se detallan los algoritmos para grupos cíclicos en general y, posteriormente, para subgrupos cíclicos del grupo de una curva elíptica. Se definen varios criptosistemas, siendo los más relevantes ElGamal y el intercambio de claves de Diffie-Hellman.Además, se abordan los ataques al logaritmo discreto, mostrando que, en particular, el Index Calculus no afecta a los criptosistemas definidos sobre el grupo de curvas elípticas. Finalmente, se presentan algunos métodos de firmas digitales.
En el Anexo 1 se puede encontrar el código en Sage de algunos ejemplos que se verán a lo largo del capítulo 3.
Las siguientes referencias son sobre las que se basa el trabajo principalmente: [5], [18] y [9].

Capítulo 1

Preliminares

Antes de poder definir y explicar la teoría de curvas elípticas, es necesario dedicar un capítulo de introducción para hablar del espacio proyectivo y de las curvas algebraicas, cuyas definiciones son básicas para el resto del trabajo.
A lo largo del trabajo, K denotará un cuerpo, V un espacio vectorial definido sobre un cuerpo K y $\mathbb{A}^n(K)$ el espacio afín de dimensión n sobre K.
Las principales referencias usadas en este capítulo han sido [5], [6] y [8].

1.1. El espacio proyectivo

En esta sección comentaremos de manera general el espacio proyectivo.

Definición 1.1.1. *Sea V un espacio vectorial definido sobre un cuerpo K. Definimos $\sim$ en $V \setminus \{0\}$ de la forma siguiente: dados $x, y \in V \setminus \{0\}$, entonces $x \sim y$ si y sólo si $\exists \lambda \in K, \lambda \neq 0$, tal que $x = \lambda y$.*

Proposición 1.1.1. *La relación definida es una relación de equivalencia.*

Demostración. $\sim$ cumple las propiedades simétrica, reflexiva y transitiva. Sean $x, y, z \in V \setminus \{0\}$.

- Simétrica: Si $x \sim y$, por definición $\exists \lambda \in K \setminus \{0\}$ tal que $x = \lambda y$. Al ser λ un elemento no nulo del cuerpo, tiene inverso, $\lambda^{-1} \in K \setminus \{0\}$, luego $y = \lambda^{-1} x$ y por tanto $y \sim x$.
- Reflexiva: Si $x \in V \setminus \{0\}$ por ser K un cuerpo se da que $x = 1_K.x$, donde 1_K representa el neutro para el producto del cuerpo. Por tanto, $x \sim x$.
- Transitiva: Si $x \sim y$ y $y \sim z$, por definición, $\exists \lambda_1, \lambda_2 \in K \setminus \{0\}$ tales que $x = \lambda_1 y$ y $y = \lambda_2 z$. En ese caso, se tiene que $x = \lambda_1.\lambda_2.z$, es decir, si $\lambda_1.\lambda_2 = \lambda_3$ (no nulo, por no serlo los anteriores y estar en un cuerpo), $x = \lambda_3 z$, luego $x \sim z$.

□

Definición 1.1.2. *En las mismas condiciones que la definición anterior, definimos el* ***espacio proyectivo*** *de V como el cociente*

$$\mathbb{P}(V) = (V \setminus \{0\})/\sim .$$

La dimensión de este espacio se define como $\dim(\mathbb{P}(V)) = \dim(V) - 1$.
Cada una de las clases de equivalencia se denomina punto proyectivo.
Como caso particular, si $V = K^{n+1}$, llamaremos el espacio proyectivo sobre K de dimensión n a $\mathbb{P}^n(K) = \mathbb{P}(K^{n+1})$.

De manera más intuitiva, los puntos del espacio proyectivo $\mathbb{P}(V)$ pueden considerarse como las rectas vectoriales de V, ya que hay una biyección natural entre los puntos proyectivos y la recta generada por alguno de sus representantes, que son vectores en $V \setminus \{0\}$. Podemos definir los subespacios proyectivos de la siguiente manera:

Definición 1.1.3. *Dado un espacio vectorial V sobre K, un* ***subespacio de*** *$\mathbb{P}(V)$ es un espacio proyectivo de la forma $\mathbb{P}(W)$, donde W es un subespacio K-lineal de V. Es decir, $\mathbb{P}(W) = \{[w]/w \in W \setminus \{0\}\}$, donde $[w]$ es la clase de w con respecto a la relación $\sim$ (nótese que $\sim$ es la relación anterior en V, pero si $w \in W$ y $v \sim w$, entonces $v \in W$, por lo que $\sim$ se puede restringir a W).*

Nota. *A lo largo del trabajo, se denotará como $[X_0 : X_1 : \cdots : X_n]$ a la clase de equivalencia de $(X_0, X_1, \cdots, X_n)$ respecto a la relación de equivalencia $\sim$. Así, para un espacio proyectivo $\mathbb{P}^n(K)$, sus puntos serán de la forma $[X_0 : X_1 : \cdots : X_n]$.*

Para este trabajo, nos interesa particularmente el plano proyectivo: $\mathbb{P}^2(K) = \mathbb{P}(K^3)$. Para este caso, se denotarán sus puntos como $[X : Y : Z]$.
Un aspecto particularmente interesante en esta teoría es la inmersión del espacio afín en el espacio proyectivo. Podemos identificar de manera no única al plano afín $\mathbb{A}^2(K)$ con un subconjunto de puntos de $\mathbb{P}^2(K)$. Detallamos una manera en la que podemos realizar esta identificación, primero de manera general:
Sea $\phi : \mathbb{A}^n(K) \longrightarrow \mathbb{P}^n(K)$ la aplicación que lleva los puntos del espacio afín a puntos del proyectivo de la siguiente manera:

$$\phi((x_1, \cdots, x_n)) = [x_1 : \cdots : x_n : 1].$$

Esta aplicación es inyectiva, ya que si $[x_1 : \cdots : x_n : 1] = [y_1 : \cdots : y_n : 1]$, entonces existiría $\lambda \neq 0$ tal que $(x_1, \cdots, x_n, 1) = \lambda(y_1, \cdots, y_n, 1)$, y por tanto, $\lambda = 1$ y $(x_1, \cdots, x_n) = (y_1, \cdots, y_n)$.
Usando esta misma aplicación, definimos el hiperplano del infinito como el conjunto de puntos de $\mathbb{P}^n(K)$ tales que su última coordenada es 0 (los puntos de la forma $[x_0 : \cdots : x_n : 0]$). Estos puntos no están en la imagen de la aplicación ϕ.
Para nuestro trabajo, dado un punto en $\mathbb{P}^2(K), [X : Y : 1]$, lo identificaremos con (X, Y). En este contexto, consideraremos como hiperplano del infinito $H = \mathbb{P}(W)$, donde $W = \{(x, y, z)/z = 0\}$.
Por último, vamos a definir qué es un polinomio homogéneo y los procedimientos de homogenización y deshomogeneización de polinomios.

Definición 1.1.4. *Diremos que $F \in K[X_1, \cdots, X_n]$ es un **polinomio homogéneo** de grado $m \in \mathbb{N}$, si podemos expresarlo como suma de monomios del mismo grado, es decir, un polinomio de la forma: $F(x_1, \cdots, x_n) = \sum_i a_i x_1^{m_{i,1}} \cdot \cdots \cdot x_n^{m_{i,n}}$ con $a_i \in K$, donde $\sum_{j=1}^{n} m_{i,j} = m$, $\forall i$.*

El interés de este tipo de polinomios radica en lo siguiente: si F es un polinomio homogéneo y $(x, y, z) = \lambda(u, v, w)$, entonces $F(x, y, z) = 0$ si, y solo si, $F(u, v, w) = 0$ (esto es, porque $F(x, y, z) = F(\lambda(u, v, w)) = \lambda^m F(u, v, w)$, con m el grado de F). En particular, que un punto proyectivo sea cero de F no depende del representante. En la siguiente sección vamos a definir las curvas algebraicas considerando conjuntos de ceros proyectivos de polinomios homogéneos, por lo que no van a depender del vector que se elija como representante de un punto proyectivo.
En la siguiente sección utilizaremos la homogeneización para relacionar una curva afín con su versión proyectiva. De manera intuitiva, la homogeneización de un polinomio consiste en añadir una variable extra para que todos los monomios tengan el mismo grado. Veamos la definición formal:

Definición 1.1.5. *Sea $F(x_1, \cdots, x_n) \in K[X_1, \cdots, X_n]$, un polinomio de grado m. Definimos la **homogeneización** de F como el polinomio $F_h(x_0, \cdots, x_n) = x_0^m F(\frac{x_1}{x_0}, \cdots, \frac{x_n}{x_0}) \in K[X_0, X_1, \cdots, X_n]$.*

Proposición 1.1.2. *Dado un polinomio $F \in K[X_1, \cdots, X_n]$ (no necesariamente homogéneo), su homogeneización F_h es un polinomio homogéneo.*

Demostración. Sea $F(x_1, \cdots, x_n)$ un polinomio de grado m no necesariamente homogéneo. Podemos escribir F como $F = \sum_i a_i x_1^{m_{i,1}} \cdot \cdots \cdot x_n^{m_{i,n}}$ con $a_i \in K$, donde lo único que podemos asegurar es que $\sum_j m_{i,j} \leq m$ para cada monomio i. Utilizando la definición, obtenemos el polinomio homogeneizado $F_h(x_0, \cdots, x_n) = \sum_i a_i x_1^{m_{i,1}} \cdot \cdots \cdot x_n^{m_{i,n}} \cdot x_0^{m - \sum_j m_{i,j}}$ con $a_i \in K$. Entonces, para cada monomio i, tenemos que su grado es $\sum_j m_{i,j} + m - \sum_j m_{i,j} = m$.

□

Para deshomogeneizar un polinomio homogeneizado, basta con dar el valor 1 a la variable extra, en nuestro caso, $x_0 = 1$. Es decir, $F(x_1, \cdots, x_n) = F_h(1, x_1, \cdots, x_n)$.
Podemos escribir la homogeneización y la deshomogenización en forma de aplicaciones y dar una relación entre ellas:

Proposición 1.1.3. *Sea h la aplicación definida por $h : K[X_1, \ldots, X_n] \longrightarrow K[X_0, X_1, \ldots, X_n]_h$, donde el segundo conjunto es el conjunto de polinomios homogéneos, dada de la siguiente manera:*

1. *Si $F = 0$, entonces $h(F) = 0$.*
2. *Si $F \neq 0$ de grado m, y F es de la forma: $F(x_1, \cdots, x_n) = \sum_i a_i x_1^{m_{i,1}} \cdot \cdots \cdot x_n^{m_{i,n}}$ con $a_i \in K$, entonces $h(F) = F_h$.*

Entonces, la aplicación $d : K[X_0, X_1, \ldots, X_n]_h \longrightarrow K[X_1, \ldots, X_n]$ definida por $d(F) = F(1, x_1, \cdots, x_n)$ cumple que $d \circ h = Id$ y $h \circ d = Id$.

Demostración. Veamos que $d \circ h = Id$.
Sea $F \in K[X_1, \cdots, X_n]$, polinomio no nulo. Tenemos entonces que :

$$h(F)(x_0, x_1, \cdots, x_n) = x_0^m F\left(\frac{x_1}{x_0}, \cdots, \frac{x_n}{x_0}\right).$$

Aplicando ahora d a $h(F)$, obtenemos:

$$d(h(F)) = h(F)(1, x_1, \cdots, x_n) = 1^m F\left(\frac{x_1}{1}, \cdots, \frac{x_n}{1}\right) = F(x_1, \cdots, x_n).$$

Veamos igualmente que $h \circ d = Id$.
Sea $F \in K[X_0, X_1, \cdots, X_n]$, polinomio no nulo y homogéneo, de la forma $\sum_i a_i x_0^{m-\sum_{j=1}^n m_{i,j}} x_1^{m_{i,1}} \cdots x_n^{m_{i,n}}$. Tenemos que:

$$d(F) = F(1, x_0, \ldots, x_n) = \sum_i a_i x_1^{m_{i,1}} \cdots x_n^{m_{i,n}}.$$

Aplicando ahora h a $d(F)$, tenemos que

$$h(d(F))) = h(\sum_i a_i x_1^{m_{i,1}} \cdots x_n^{m_{i,n}}) = \sum_i a_i x_0^{m-\sum_{j=1}^n m_{i,j}} x_1^{m_{i,1}} \cdots x_n^{m_{i,n}} = F.$$

□

Ilustramos con un pequeño ejemplo cómo homogeneizar y deshomogeneizar un polinomio:

Ejemplo 1.1.1. *Sea $F(x, y) = x^4 + 5x^2y^2 + xy + y$ un polinomio de grado 4 en dos variables. Utilizando lo visto, su polinomio homogeneizado será $F_h = x^4 + 5x^2y^2 + xyz^2 + yz^3$, un polinomio de grado 4 en tres variables.*
Para deshomogeneizarlo basta con hacer $z = 1$, y obtenemos el polinomio original F.

Veamos una propiedad interesante de la homogeneización:

Corolario 1.1.1. *Dados $F, G \in K[X_1, \cdots, X_n]$ no nulos, tenemos que $h(FG) = h(F)h(G)$.*

Demostración. Sean F y G dos polinomios no nulos de grados M_1 y M_2 respectivamente, que escribimos como $F(x_1, \ldots, x_n) = \sum_i a_i x_1^{m_{i,1}} \cdots x_n^{m_{i,n}}$ y $G(x_1, \ldots, x_n) = \sum_j b_j x_1^{k_{j,1}} \cdots x_n^{k_{j,n}}$.
Utilizando la aplicación definida en la proposición 1.1.3, tenemos que $h(F) = \sum_i a_i X_0^{m_i} X_1^{m_{i,1}} \cdots X_n^{m_{i,n}}$ y $h(G) = \sum_j b_j X_0^{k_j} X_1^{k_{j,1}} \cdots X_n^{k_{j,n}}$, donde $m_i = M_1 - \sum_{j=1}^n m_{i,j}$ y $k_i = M_2 - \sum_{j=1}^n k_{i,j}$.
De igual manera, desarrollando el producto FG, obtenemos que $FG(x_1, \ldots, x_n) = \sum_{i,j} a_i b_j x_1^{m_{i,1}+k_{j,1}} \cdots x_n^{m_{i,n}+k_{j,n}}$, y su homogeneización es

$$h(FG) = \sum_{i,j} a_i b_j X_0^{m_i+k_j} X_1^{m_{i,1}+k_{j,1}} \cdots X_n^{m_{i,n}+k_{j,n}}.$$

Por último, se calcula el producto de los homogeneizados y comprobamos que es igual al homogeneizado del producto.

$$h(F)h(G) = \sum_{i,j} a_i b_j X_0^{m_i+k_j} X_1^{m_{i,1}+k_{j,1}} \cdot X_n^{m_{i,n}+k_{j,n}}.$$

Como efectivamente $h(FG) = h(F)h(G)$, se da por finalizada la prueba. □

1.2. Curvas algebraicas

En esta sección introduciremos los conceptos generales de curvas algebraicas, teoría desarrollada principalmente durante el siglo XIX. Lo comentado en esta sección sirve de base teórica para el capítulo de curvas elípticas.
Cabe destacar que en esta sección los cuerpos sobre los que se definen las curvas los consideraremos algebraicamente cerrados por simplificar algunos resultados. Sin embargo, la mayor parte de esta teoría es igualmente válida para cuerpos finitos, lo que nos permitirá definir curvas elípticas sobre cuerpos finitos.

Definición 1.2.1. *1. Sea $F \in K[X_1, \cdots, X_n]$ un polinomio, y $P = (p_1, \cdots, p_n)$ un punto del n- espacio afín de K. Se dice que P es un **cero** de F si y sólo si $F(p_1, \cdots, p_n) = 0$.*

*2. Si F es un polinomio en K no constante, el conjunto de ceros del polinomio, $V(F) = \{P \in \mathbb{A}^n(K)/F(P) = 0\}$ se denomina como **hipersuperficie** definida por F.*
De manera general, si tomamos S un conjunto de polinomios en $K[x_1, \ldots, x_n]$, podemos hablar de los ceros del conjunto S, $V(S) = \{P \in \mathbb{A}^n(K)/F(P) = 0, \forall F \in S\}$, o lo que es lo mismo, $V(S)$ es la intersección del conjunto de puntos que son cero para cada $F \in S$.

Con todo esto, estamos preparados para introducir la definición de conjunto algebraico:

Definición 1.2.2. *Dado $X \in \mathbb{A}^n(K)$, diremos que X es un **conjunto algebraico** si existe un conjunto de polinomios S en $K[x_1, \ldots, x_n]$ tal que $X = V(S)$.*

Vamos a introducir el concepto de conjunto algebraico irreducible. Para ello, definiremos el ideal de un subconjunto de $\mathbb{A}^n(K)$, daremos una serie de propiedades de los conjuntos algebraicos y por último, diremos cuándo un conjunto algebraico es reducible.

Definición 1.2.3. *Sea $X \subset \mathbb{A}^n(K)$. El conjunto de polinomios que se anulan en X forma un ideal en $K[x_1, \ldots, x_n]$ al que llamaremos **ideal de** X, denotado por $I(X)$. Explícitamente,*

$$I(X) = \{F \in K[x_1, \ldots, x_n]/F(P) = 0, \forall P \in X\}.$$

Proposición 1.2.1. *1. Si $I, J \subseteq K[x_1, \ldots, x_n]$ tales que $I \subset J$, entonces $V(J) \subset V(I)$.*

2. *Si I es un ideal de $K[x_1, \ldots, x_n]$ generado por S, entonces $V(S) = V(I)$. Como consecuencia, todo conjunto algebraico es igual a $V(I)$ para algún ideal I.*

3. *Dados $F_1, F_2 \in I(V)$, se tiene que $V(F_1F_2) = V(F_1) \cup V(F_2)$.*

4. *Sean V_1, V_2 dos conjuntos algebraicos. Entonces, $I(V_1 \cup V_2) = I(V_1) \cap I(V_2)$.*

5. *Si V es un conjunto algebraico tal que $V = V(S)$, entonces $V(I(V)) = V$.*

Demostración. 1. Utilizando la definición,

$$V(J) = \bigcap_{F \in J} V(F) \subset \bigcap_{G \in I} V(G) = V(I),$$

donde la contención se da ya que $I \subset J$ implica que los polinomios que definen I también están en J.

2. Como $S \subseteq I = \langle S \rangle$, por el punto anterior tenemos que $V(I) \subset V(S)$.
Para ver la otra implicación, sea $x \in V(S)$ y $F \in I$, y veamos que también $x \in V(F)$. Si escribimos $F = g_1F_1 + \cdots + g_mF_m$, con $F_i \in S, g_i \in K[x_1, \ldots, x_n]$, entonces $F(x) = 0$, ya que al ser $x \in V(S)$, cada sumando se anula por $F_i(x) = 0$.

3. Veamos la doble inclusión:
$V(F_1F_2) \subseteq V(F_1) \cup V(F_2)$
Sea $P \in V(F_1F_2)$, luego, utilizando la definición, sabemos que $F_1(P)F_2(P) = 0$. Si $F_1(P) = 0$, entonces $P \in V(F_1)$. Análogamente, si $F_2(P) = 0$, entonces $P \in V(F_2)$. En cualquier caso, tenemos que $P \in V(F_1) \cup V(F_2)$, con lo que concluimos la demostración de la primera inclusión.
$V(F_1) \cup V(F_2) \subseteq V(F_1F_2)$
Sea $P \in V(F_1) \cup V(F_2)$. Tenemos entonces dos opciones:

   - Si $P \in V(F_1)$, entonces $F_1(P) = 0$ y el producto $F_1(P)F_2(P)$ también se anula. Luego $P \in V(F_1F_2)$.
   - Si $P \in V(F_2)$, entonces $F_2(P) = 0$ y el producto $F_1(P)F_2(P)$ también se anula. Luego $P \in V(F_1F_2)$.

   En cualquier caso, $P \in V(F_1F_2)$.

4. Veamos la doble contención:

   - $I(V_1 \cup V_2) \subseteq I(V_1) \cap I(V_2)$.
   Supongamos que $F \in I(V_1 \cup V_2)$, esto es $F(P) = 0, \forall P \in V_1 \cup V_2$. Entonces, F se anula para todo punto de V_1 (resp. V_2), por lo que $F \in I(V_1)$ (resp. $F \in I(V_2)$) y $F(P) = 0, \forall P \in V_1$ (resp. $\forall P \in V_2$)).Por tanto, $F \in I(V_1)$ (resp. $F \in I(V_2)$), por lo que F está en su intersección.

- $I(V_1) \cap I(V_2) \subseteq I(V_1 \cup V_2)$.
  Supongamos que $F \in I(V_1) \cap I(V_2)$, esto es $F(P) = 0, \forall P \in V_1$ y $F(P) = 0, \forall P \in V_2$. Es decir, $F(P) = 0, \forall P \in V_1 \cup V_2$. Por tanto, $F \in I(V_1 \cup V_2)$.

5. Demostramos la doble contención:

   - $V \subseteq V(I(V))$:
     Dado que construimos $I(V)$ como los polinomios que anulan a los puntos de V, todos los polinomios en $I(V)$ se anulan en particular en V, luego $V \subset V(I(V))$.
   - $V(I(V)) \subseteq V$:
     Si $F \in S$ es un polinomio, tenemos que $\forall P \in V(S)$, $F(P) = 0$, por construcción, luego $F \in I(V(S))$. Por tanto, $S \subseteq I(V(S))$ y entonces $V(I(V(S))) \subseteq V(S)$, como queríamos probar.

□

Definición 1.2.4. *Un conjunto algebraico, $V \subset \mathbb{A}^n(K)$ es **reducible** si y sólo si $\exists V_1, V_2 \subset \mathbb{A}^n$, conjuntos algebraicos no vacíos y distintos a V, tales que $V = V_1 \cup V_2$.*
*V es **irreducible** si no es reducible.*

Proposición 1.2.2. *Un conjunto algebraico V es irreducible si y sólo si $I(V)$ es primo.*

Demostración. Veamos las dos implicaciones, utilizando las propiedades vistas en la proposición 1.2.1 :
$\Longrightarrow$) Supongamos que V es irreducible. Sean $F_1, F_2 \in K[x_1, \ldots, x_n]$ tales que $F_1F_2 \in I(V)$. Entonces, como $\langle F_1F_2 \rangle \subseteq I(V)$ tenemos que $V(F_1F_2) = V(F_1) \cup V(F_2)$, y por lo tanto, $V \subseteq V(F_1) \cup V(F_2)$. Debido a que V es irreducible, escribiendo $V = (V(F_1) \cup V(F_2)) \cap V = (V(F_1) \cap V) \cup (V(F_2) \cap V)$, debe cumplirse que $V \subseteq V(F_1)$ o $V \subseteq V(F_2)$, lo cual implica que $F_1 \in I(V)$ o $F_2 \in I(V)$. Por lo tanto, $I(V)$ es un ideal primo.
$\Longleftarrow$) Supongamos que $I(V)$ es primo. Razonemos por reducción al absurdo: si V es reducible, por definición, $\exists V_1, V_2 \subseteq \mathbb{A}$, conjuntos algebraicos no vacíos y distintos a V, tales que $V = V_1 \cup V_2$. Entonces. $I(V) = I(V_1) \cap I(V_2)$. Notemos que $I(V) \neq I(V_1)$, ya que sino tendríamos $V = V_1$.
Sea $F_1 \in I(V_1) \setminus I(V)$. Para todo $F_2 \in V(V_2), F_1F_2 \in I(V_1) \cap I(V_2) = I(V)$, y como $F_1 \notin I(V)$, $F_2 \in I(V)$, por ser este primo. Entonces, $I(V) = I(V_2)$, lo que implica $V = V_2$, lo cual es absurdo. □

Definición 1.2.5. *Llamamos **variedades afines** a los conjuntos algebraicos afines irreducibles.*

Volviendo al objetivo del trabajo, definiremos las curvas planas afines y las curvas planas afines con multiplicidad.

Definición 1.2.6. *Se denomina **curva plana afín** a una hipersuperficie $V(F)$ de $\mathbb{A}^2(K)$.*

Definamos la siguiente relación de equivalencia que vamos a usar:

Definición 1.2.7. *Sean $F, G \in K[X,Y]$ dos polinomios (no constantes). Decimos que F y G son equivalentes módulo $\sim$ si existe $\lambda \in K \setminus \{0\}$ tal que $F = \lambda G$.*

Definición 1.2.8. *Una **curva plana afín con multiplicidad** (definida sobre K) es un polinomio no constante, $F \in K[X,Y]$ módulo la relación de equivalencia anterior, $\sim$.*
A la curva se la denominará igual que al polinomio, F.

La necesidad de esta definición es poder contar puntos con multiplicidad de manera más sencilla, y lo necesitaremos para probar la asociatividad de la ley de grupo de una curva elíptica en la sección 2.2.1.
Cabe destacar que todas las propiedades y definiciones que se dan respecto a las curvas planas están inequívocamente definidas, ya que si se toma otro polinomio equivalente para definir la curva, no cambiará sus propiedades al multiplicarlo por $\lambda \in K \setminus \{0\}$.

Nota. *Si $F = \lambda G$, entonces $V(F) = V(G)$; es decir, una curva plana afín con multiplicidad tiene asociada una única curva plana afín, que es su conjunto de puntos en $\mathbb{A}^n(K)$. Sin embargo, una curva plana afín en $\mathbb{A}^n(K)$ puede tener asociadas dos curvas planas afines con multiplicidad (por ejemplo, $V(F)$ tiene asociadas F y F^2 como curvas planas afines con multiplicidad, ya que en el segundo caso se cuentan los puntos con el doble de multiplicidad).*

Veamos una serie de definiciones asociadas a las curvas planas afines con multiplicidad.

Definición 1.2.9.

- *El **grado** de una curva plana afín con multiplicidad es el grado del polinomio F que la define.*
- *Diremos que una curva plana afín con multiplicidad es **irreducible** si lo es el polinomio que la define. Sino, diremos que es **reducible**.*
- *Si F es el polinomio que define a una curva plana afín con multiplicidad, y $F = F_1^{e_1} \cdots F_k^{e_k}$ es su descomposición en polinomios en K, irreducibles, entonces también será la descomposición irreducible de la curva F que define, y llamaremos a las curvas planas afines $F_1, \cdots, F_k$ las componentes irreducibles de F. A los $\{e_i \in \mathbb{N} / i = 1, 2, \ldots, k\}$ se les denomina multiplicidades.*

Nota. *En el caso de las curvas planas afines, el grado de la curva no es el grado del polinomio que la define: por ejemplo, F y F^2 generan la misma curva ($V(F) = V(F^2)$), pero F y F^2 no tienen el mismo grado.*

Veamos un pequeño ejemplo:

Ejemplo 1.2.1. *Sea $F = x^2 + y^2 - 1$ definido en $\mathbb{R}$. Entonces, F puede ser considerada como una curva plana afín con multiplicidad. Además, al ser un*

polinomio irreducible, también lo es la curva.
Si tomamos $G = 2x^2 + 2y^2 - 2$, en el mismo cuerpo, tenemos que F y G generan la misma curva con multiplicidad, ya que $G = 2F$. En particular, ambos polinomios (y la curva) tienen los mismos puntos en $\mathbb{A}^2(\mathbb{R})$.
Por otro lado, también tenemos que $V(F) = V(F^2) = V((x^2 + y^2 - 1)^2)$, por lo que es la misma curva plana afín, pero F y F^2 no definen la misma curva con multiplicidad, ya que tienen distinto grado (de hecho, F^2 define una curva plana afín con el doble de multiplicidades que F).

Nota. *Un polinomio puede no tener puntos en los que se anule. Sin embargo, la curva sigue estando definida, aunque su conjunto de puntos también sea vacío.*

Veamos a continuación algunas definiciones que caracterizan a estas curvas:

Definición 1.2.10. *Sea F una curva plana afín con multiplicidad.*

1. *Dado P un punto en F, diremos que P es un **punto simple** si alguna de las derivadas parciales de F sobre P es distinta de cero.*
   *Si un punto no es simple, diremos que es un **punto singular**.*

2. *Se dirá que una curva es **no singular** si todos sus puntos son simples.*

Para finalizar esta sección, y poder adentrarnos en el capítulo de las curvas elípticas, introduciremos los conceptos más generales de las curvas proyectivas. De manera similar a lo realizado para curvas planas afines, definimos una curva proyectiva.

Definición 1.2.11. *Una **curva plana proyectiva con multiplicidad** (sobre un cuerpo K) es un polinomio homogéneo no constante, $F \in K[X, Y, Z]$ módulo $\sim$, donde $\sim$ es la relación de equivalencia de la definición 1.2.7.*
El conjunto $V(F) = \{P \in \mathbb{P}^2(K)/F(P) = 0\}$ es la curva plana proyectiva asociada a F.

Los conceptos de grado de una curva proyectiva, irreductibilidad y multiplicidades son iguales a los vistos para curva afines.
Las curvas afines y las proyectivas están relacionadas: dada F una curva proyectiva, el conjunto de puntos P tales que $P \in V(F) \cap \mathbb{A}^2$, pertenecen a la curva plana afín que corresponde a la deshomogeneización del polinomio F.
Los puntos del infinito de F serán de la forma $\{P = [x : y : 0]/F(P) = 0, P \in \mathbb{P}^2(K)\}$.

Ejemplo 1.2.2. *Sea $F_h = xy^4 + xz^4 + yz^4$. Entonces, F_h es una curva plana proyectiva con multiplicidad, y su versión afín será $F = xy^4 + x + y$.*

Capítulo 2

Curvas elípticas

A lo largo de este capítulo vamos a desarrollar la teoría matemática asociada a las curvas elípticas. Se divide el capítulo en tres secciones: en la primera vamos a definir y dar las principales propiedades de las curvas elípticas, vistas de manera general. A continuación, detallaremos la ley de grupo. Por último, con el objetivo de preparar el capítulo de criptografía sobre curvas elípticas, nos centraremos en curvas elípticas sobre cuerpos finitos.
Las principales referencias de este capítulo han sido [14], [15] y [7].

2.1. Curvas elípticas: primeros resultados

Definición 2.1.1. *Una **curva elíptica** E sobre K es una curva plana proyectiva (sobre K) irreducible y no singular de grado 3.*

Nota. *En el resto del trabajo, utilizaremos la notación $E(K)$ para denotar los puntos de la curva elíptica que se defina sobre el cuerpo K.*

Este tipo de curvas admiten una representación que será la que utilicemos durante el resto del capítulo. Esta representación se denomina forma o ecuación de Weierstrass:

Definición 2.1.2. *Una curva elíptica en **forma de Weierstrass generalizada** sobre el cuerpo K es la curva de ecuación: $y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$, con $a_i \in K$.*
El conjunto de puntos $(x, y) \in K^2$ que satisfacen la ecuación de Weierstrass, junto con el punto del infinito, $O = [0 : 1 : 0]$ son los puntos de la curva.

Proposición 2.1.1. *Si consideramos E una curva elíptica en forma de Weierstrass generalizada, el punto $[0 : 1 : 0]$ es el único punto del infinito de la curva.*

Demostración. En primer lugar, homogeneizamos la ecuación de Weierstrass, y obtenemos:

$$y^2z + a_1xyz + a_3yz^2 = x^3 + a_2x^2z + a_4xz^2 + a_6z^3.$$

Evaluamos ahora el hiperplano del infinito, que según lo visto en el capítulo anterior, es de la forma $[x : y : 0]$, por lo que sustituyendo $z = 0$ en la ecuación homogeneizada, resulta:

$$y^2 \cdot 0 + a_1xy \cdot 0 + a_3y \cdot 0^2 = x^3 + a_2x^2 \cdot 0 + a_4x \cdot 0^2 + a_6 \cdot 0^3.$$

Con lo cual, $0 = x^3$ y resulta $x = 0$. Al ser coordenadas proyectivas, tomamos la coordenada $y = 1$ ya que representa a cualquier punto de la forma $[0 : y : 0]$. Por tanto, el punto del infinito en este caso es $[0 : 1 : 0]$. □

Proposición 2.1.2. *Sea K un cuerpo de característica distinta de 2 y de 3. Tras un cambio afín de variable, una curva elíptica en forma de Weierstrass generalizada, $(E(K))$, es el conjunto de soluciones $(X, Y) \in K^2$ de la ecuación $f(X, Y) = Y^2 - X^3 - aX - b$, donde $a, b \in K$. Esta forma se denomina* ***curva de Weierstrass simplificada****.*

Demostración. Veamos que ambas expresiones de la curva de Weierstrass son equivalentes, y que podemos transformar una ecuación en la otra mediante un cambio de variable afín:
Partimos de la forma no simplificada:

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6.$$

Vamos a simplificar la forma, buscando completar cuadrados:

$$\left(y + \frac{a_1x}{2} + \frac{a_3}{2}\right)^2 = x^3 + \left(a_2 + \frac{a_1^2}{4}\right)x^2 + \left(a_4 + \frac{a_1a_3}{2}\right)x + \left(a_6 + \frac{a_3^2}{4}\right).$$

Realizamos el cambio de variable $Y = y + \frac{a_1x}{2} + \frac{a_3}{2}$, y sustituimos en la ecuación original:

$$Y^2 = x^3 + \left(a_2 + \frac{a_1^2}{4}\right)x^2 + \left(a_4 + \frac{a_1a_3}{2}\right)x + \left(a_6 + \frac{a_3^2}{4}\right).$$

Renombrando los coeficientes como $A_2 = a_2 + \frac{a_1^2}{4}$, $A_4 = a_4 + \frac{a_1a_3}{2}$ y $A_6 = a_6 + \frac{a_3^2}{4}$, obtenemos $Y^2 = x^3 + A_2x^2 + A_4x + A_6$.
Ahora, mediante el cambio de variable $X = x + \frac{A_2}{3}$, sustituyendo y simplificando en la ecuación anterior, obtenemos:

$$Y^2 = X^3 + \left(A_4 - \frac{A_2^2}{3}\right)X + \left(-\frac{A_2^3}{27} + \frac{A_2^3}{9} - \frac{A_4A_2}{3} + A_6\right).$$

Definiendo $A = A_4 - \frac{A_2^2}{3}$ y $B = -\frac{A_2^3}{27} + \frac{A_2^3}{9} - \frac{A_4A_2}{3} + A_6$, con lo que llegamos a la ecuación de Weierstrass simplificada.
Al estar todas las transformaciones afines bien definidas ya que los cambios de variable realizados tienen una inversa bien definida en el cuerpo, esta transformación entre la ecuación de Weierstrass generalizada y la simplificada es una biyección.

□

Siempre que el cuerpo en el que estemos trabajando lo admita, utilizaremos la ecuación de Weierstrass simplificada para definir curvas elípticas.

Definición 2.1.3. *Sea E una curva elíptica definida sobre un cuerpo K con una ecuación de Weierstrass simplificada dada por: $y^2 = x^3 + Ax + B$, con $A, B \in K$.*

La forma proyectiva de la ecuación de Weierstrass de E en $\mathbb{P}^2(K)$ se obtiene al pasar de coordenadas afines (x, y) a $[X : Y : Z]$, que son las coordenadas homogéneas de un punto en el plano proyectivo. La ecuación proyectiva correspondiente es: $Y^2Z = X^3 + AXZ^2 + BZ^3$, donde X, Y, Z son las coordenadas homogéneas, y esta ecuación es la ***representación proyectiva de la curva elíptica*** *E.*

Veamos a continuación dos ejemplos de curvas elípticas sobre $\mathbb{R}$:

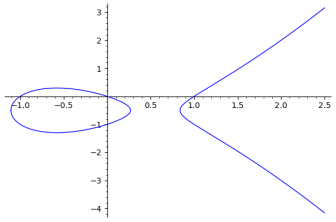

Figura 2.1: Curva elíptica $y^2 + y = x^3 - x$

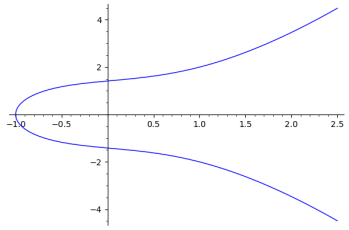

Figura 2.2: Curva elíptica $y^2 = x^3 + x + 2$

Definición 2.1.4. *Definimos el* ***discriminante*** *de una curva elíptica (en forma generalizada de Weierstrass), $\triangle$, por $\triangle = -\delta_1^2\delta_4 - 8\delta_2^3 - 27\delta_3^2 + 9\delta_1\delta_2\delta_3$, donde $\delta_1 = a_1^2 + 4a_4, \delta_2 = 2a_4 + a_1a_3, \delta_3 = a_3^2 + 4a_6, \delta_4 = a_1^2a_6 + 4a_2a_6 - a_1a_3a_4 + a_2a_3^3 - a_4^2$.*
Si la característica del cuerpo es distinta de 2 y 3, como hemos visto, podemos utilizar la ecuación simplificada de Weierestrass, y simplificamos el discriminante: $\triangle = -16(4A^3 + 27B^2)$.

En lo que resta del trabajo, denotaremos por E una curva elíptica si no hay dudas del cuerpo en el que se está trabajando. Además, a no ser que se

especifique lo contrario, trabajaremos en cuerpos con característica distinta de 2 y 3. Por tanto, utilizaremos la ecuación en forma de Weierstrass simplificada para representar las curvas.

Proposición 2.1.3. *Si E es una curva dada por la forma simplificada de Weierstrass, es una curva elíptica (es decir, es irreducible y no singular de grado 3) si y sólo si $\triangle \neq 0$.*

Demostración. Dada una curva en forma simplificada de Weierstrass, $E(K) = Y^2 - X^3 - AX - B$, ésta es una curva elíptica si y sólo si es no singular.
Veamos que una curva en forma simplificada de Weierstrass es singular si y sólo si $\triangle = 0$.
Recordamos que un punto $P = (x_0, y_0)$ es singular en una curva F si las derivadas parciales de F en P son nulas.
E una curva singular, si y sólo si, existe un punto singular, es decir, $P = (x_0, y_0) \in K^2$ tal que $2y_0 = 0$ y $-3x_0^2 - A = 0$, es decir, $y_0 = 0$ y $x_0^2 = -\frac{A}{3}$.
Sustituyendo en la forma original, obtenemos $0 = -x_0(-\frac{A}{3}) - Ax_0 - B$, por lo que $x_0^2 = \frac{-A}{3}$ y $x_0^2 = \frac{9B^2}{4A^2}$.
Juntando ambas igualdades, la curva es singular si y sólo si $4A^3 + 27B^2 = 0$.
□

Proposición 2.1.4. *La condición $\triangle \neq 0$ es equivalente a que el polinomio $X^3 + AX + B$ no tenga raíces de grado superior a uno, es decir, que sus tres raíces sean distintas.*

Demostración. Consideremos el polinomio $X^3 + AX + B$ y supongamos que se puede factorizar como:

$$X^3 + AX + B = (X - z_1)(X - z_2)(X - z_3),$$

donde z_1, z_2, z_3 son las raíces del polinomio. Entonces $4A^3 + 27B^2 \neq 0$ si y sólo si z_1, z_2 y z_3 son distintos entre sí.
Tenemos la igualdad

$$X^3 + AX + B = X^3 - (z_1 + z_2 + z_3)X^2 + (z_1z_2 + z_1z_3 + z_2z_3)X - z_1z_2z_3.$$

Comparando los coeficientes de cada término en X, obtenemos las siguientes relaciones:

- Para X^0, $z_1z_2z_3 = B$,
- Para X, $z_1z_2 + z_1z_3 + z_2z_3 = A$,
- Para X^2, $z_1 + z_2 + z_3 = 0$.

Vamos a continuar la prueba por contrarrecíproco: supongamos que hay dos raíces iguales, $z_1 = z_2$. Sustituyendo en las tres igualdades anteriores, obtenemos:

$$2z_1 + z_3 = 0, \quad z_1^2 + 2z_1z_3 = A, \quad z_1^2z_3 = B.$$

Resolviendo este sistema obtenemos $z_3 = -2z_1$, y sustituyendo en las expresiones para A y B, tenemos:

$$A = z_1^2 - 4z_1^2, \quad B = -2z_1^3.$$

Entonces, obtenemos:

$$A = -3z_1^2, \quad B = -2z_1^3.$$

Ahora calculamos el discriminante:

$$\Delta = 4A^3 + 27B^2 = 4(-3z_1^2)^3 + 27(-2z_1^3)^2 = -108z_1^6 + 108z_1^6 = 0.$$

Por lo tanto, si hay raíces iguales, el discriminante es nulo.

Vamos a demostrar el otro sentido: supongamos que el discriminante es nulo, es decir, $4A^3+27B^2 = 0$. Sustituyendo los valores de las igualdades del principio de la prueba y operando:

$$\begin{aligned} 4A^3 + 27B^2 = &(4z_2^3 + 12z_3z_2^2 + 12z_3^2z_2 + 4z_3^3)z_1^3 \\ &+ (12z_3z_2^3 + 51z_3^2z_2^2 + 12z_3^3z_2)z_1^2 \\ &+ (12z_3^2z_2^3 + 12z_3^3z_2^2)z_1 \\ &+ 4z_3^3z_2^3. \end{aligned}$$

Sustituyendo $z_1 = -z_2 - z_3$, obtenemos que los anteriores sumandos equivalen a:

$$-4z_2^6 - 12z_3z_2^5 + 3z_3^2z_2^5 + 3z_3^2z_2^4 + 26z_3^3z_2^3 + 3z_3^4z_2^2 - 12z_3^5z_2 - 4z_3^6.$$

Como $(z_2 - z_3)^2$ es un factor común de todos los sumandos y, dado que $z_3 = -z_2 - z_3$, $(z_2 + 2z_3)^2$ es igualmente un factor de la expresión, obtenemos:

$$4A^3 + 27B^2 = -(z_2 - z_3)^2(z_2 + 2z_3)^2(z_3 + 2z_2)^2.$$

Finalmente, teniendo en cuenta que $z_1 + z_2 + z_3 = 0$, el discriminante es nulo si y sólo si $(z_2 - z_3)^2(z_1 - z_3)^2(z_1 - z_2)^2 = 0$, es decir, si alguna de sus raíces coinciden. □

2.2. Ley de grupo

El objetivo de la sección es demostrar que $(E(K), +)$ es un grupo para cierta suma $+$ que definiremos a continuación. Esta estructura nos permitirá usar las curvas elípticas en criptografía.
Definamos la operación de suma de manera geométrica.

Proposición 2.2.1. *Sean $P, Q \in E(K)$, dos puntos de una curva elíptica. Distinguimos varios casos:*

1. *Si $P \neq Q$, y $P, Q \neq O$, consideramos la recta r_{PQ}, la recta que pasa por P y Q. Veamos en qué casos existe otro punto distinto en $E(K)$, que corta con la recta r_{PQ}.*

   a) *Si $P = (x, y)$ y $Q \neq (x, -y)$, entonces P y Q tienen la primera componente distinta (ya que para x fijo, los únicos puntos en $E(K)$ que tienen de primera componente x son, $(x, y), (x, -y)$ y O; al estar usando la forma simplificada, para x fijo, el único término con y es y^2). Por tanto, el vector $\vec{PQ} = (a, b)$ tiene la primera coordenada distinta de 0. Entonces, la expresión explícita de la recta r_{PQ} es $(x + \lambda a, y + \lambda b), \lambda \in K$. Al intersecar esta recta con $E(K)$, es decir, $f(x, y) = 0$, $A(x + \lambda a) + (y + \lambda b)^2 - (x + \lambda a)^3 + B$, que tiene tres raíces de multiplicidad 1, P, Q, y otra solución $R = (x_3, y_3)$.La suma se define en este caso como:*

   $$P + Q = (x_3, -y_3).$$

   b) *Si $P = (x, y)$ y $Q = (x, -y)$, $\vec{PQ} = (0, b)$, la intersección con $E(K) \setminus O$ solo va a tener dos raíces, las conocidas, P y Q. Como la intersección tiene tres raíces en $E(K)$, entonces el tercer punto de corte sucede en el infinito (si homogeneizamos la recta $X = x_0$ queda $X - x_0 Z$, que en el infinito $Z = 0$ tiene el punto $[0 : 1 : 0]$ ya que debe cumplir $Z = 0$ por tanto $X = 0$, y justo $[0 : 1 : 0] = O$), por lo que la suma es:*

   $$P + Q = O.$$

2. *Si $P = Q$ y $P, Q \neq O$. Consideramos la recta tangente a $E(K)$ en el punto P.*

   a) *Si $P = (x, 0)$, la recta tangente es la vertical, que no corta a $E(K)$ en ningún punto afín, por lo que lo cortaría en el infinito. Esto es porque las rectas verticales cortan a E en el punto del infinito (por lo visto en el apartado b)). Por tanto la suma es:*

   $$P + Q = O.$$

   b) *Si $P = (x, y)$, con $y \neq 0$, la tangente cortará a $E(K)$ en algún punto con multiplicidad 1 (ya que corta a P con multiplicidad 2), que se calculará haciendo la intersección, $R = (x_3, y_3)$. La suma*

   $$P + Q = P + P = (x_3, -y_3).$$

3. *Si $P \neq O, Q = O$. Entonces, $P + O = P$.*

4. *Si $P = Q = O$, entonces, $P + Q = O$.*

Definición 2.2.1. *Definimos la* ***suma de puntos de una curva elíptica*** *como $P + Q$ de la proposición anterior.*

Veámoslo de manera gráfica con un ejemplo:

Ejemplo 2.2.1. *Sea* $E(\mathbb{R}) : y^2 = x^3 + x + 7$. *Utilizando Elliptic curves Points para la animación, sumamos los dos puntos marcados.* [1]

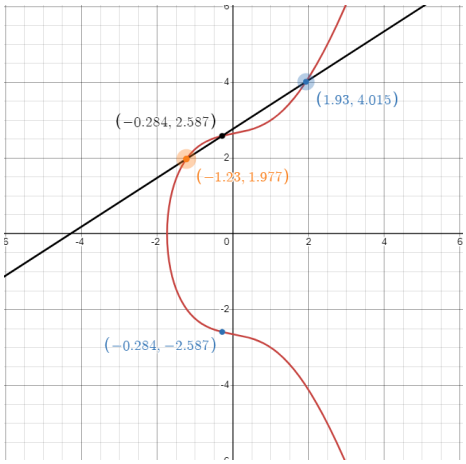

Figura 2.3: Curva elíptica $y^2 = x^3 + x + 7$

Como estamos en el caso 1a, donde ni P *(en naranja) ni* Q *(en azul) son el punto del infinito, ni simétricos entre sí, trazamos una recta que une ambos puntos, y calculamos el simétrico del tercer punto de corte de la recta con la curva. Este punto es la suma de* P *y* Q.

Teorema 2.2.1. $(E(K), +)$, *con la suma definida anteriormente es un grupo conmutativo.*

Demostración. Probemos todos los axiomas de grupos:

1. Existencia de elemento neutro: $P + O = P, \forall P \in E(K)$.
   Sea $P \in E(K)$ un punto cualquiera. Por la definición dada en la suma, $P + O = P$, para cualquier P, luego O es el elemento neutro.

2. Existencia de elemento inverso: $P + P' = O$
   El inverso de O es él mismo. Para cualquier otro $P = [x : y : 1]$, el inverso $-P = [-x : y : -1]$, y este último es igual a $-P = [x : -y : 1]$ Este punto, geométricamente, es el punto simétrico de P respecto al eje horizontal.

3. Conmutatividad: $P + Q = Q + P$, $\forall P, Q \in E(K)$
   Sean $P, Q \in E(K)$. La conmutatividad resulta de manera inmediata. ya que la recta r_{PQ} es la misma que la recta r_{QP}, y por tanto, el punto de corte afín (si existe) con $E(K)$ es el mismo. En caso contrario, sería O.

[1] https://www.desmos.com/calculator/ialhd71we3?lang=es

4. Asociatividad: $(P+Q)+R = P+(Q+R), \forall P,Q,R \in E(K)$.
   Esta propiedad se estudiará en la subsección 2.2.1.

□

Nota. *La elección del punto O no es relevante, ya que si eligiésemos otro elemento para ser el cero, O', a través de un isomorfismo $P \to P + (O' - O)$, la estructura del grupo se mantiene.*

Podemos calcular la suma de dos puntos de una curva elíptica de manera explícita según sus coordenadas:

Teorema 2.2.2. *Sea E una curva elíptica en la forma simplificada de Weierstrass: $E(K) : Y^2 = X^3 + AX + B$. Sean $P, Q \in E(K)$, dos puntos de la curva. Vamos a calcular la suma $P+Q$ de manera explícita, dependiendo de los casos:*

1. *Si $P = O$ (equivalentemente, $Q = O$), $P + Q = Q$ (equivalentemente, $P + Q = P$).*
2. *Si $P = (p_1, p_2)$, y $Q = (p_1, -p_2)$, entonces $P + Q = O$.*
3. *Si $P = (p_1, p_2)$, y $Q = (q_1, q_2)$, distintos de los casos 1 y 2 ya tratados, entonces*
   Sea $\lambda = \begin{cases} \frac{q_2 - p_2}{q_1 - p_1} & si\ P \neq Q, \\ \frac{3p_1^2 + A}{2p_2} & si\ P = Q. \end{cases}$
   Entonces, Si $P + Q = R = (r_1, -r_2)$, tenemos $r_1 = \lambda^2 - p_1 - q_1$, y $r_2 = \lambda(p_1 - r_1) - p_2$.

Demostración. Los dos primeros puntos los hemos demostrado previamente, por lo que vamos a centrarnos en el tercer caso. En primer lugar, notemos qué es λ en cada caso.
Para $P = Q$, λ queda definida como la pendiente de la tangente al punto P. Si $P \neq Q$, λ es la pendiente de la recta que pasa por P y Q.
Como la suma es el punto de corte entre $E(K)$ y la recta correspondiente (dependiendo de si estamos sumando el mismo punto o dos distintos), nos interesa escribir la ecuación de la recta correspondiente. En cualquier caso, la ecuación de la recta es:

$$Y = \lambda X + (p_2 - \lambda p_1).$$

Sustituyendo Y en la ecuación de Weierstrass, obtenemos una ecuación de tercer grado en X. Como sabemos que P y Q están en la intersección de las curvas, las primeras coordenadas, p_1 y q_1, son soluciones a esta ecuación. Como sabemos que tiene tres soluciones, buscamos la tercera:

$$(\lambda X + (p_2 - \lambda p_1))^2 = X^3 + AX + B,$$

lo que da:

$$X^3 - \lambda^2 X^2 + (A - 2\lambda(p_2 - \lambda p_1))X + (B - (p_2 - \lambda p_1)^2) = 0.$$

Esta ecuación puede factorizarse como:

$$0 = (X-p_1)(X-q_1)(X-r_1) = X^3-X^2(p_1+q_1+r_1)+X(p_1q_1+r_1q_1+r_1p_1)+p_1q_1r_1.$$

Igualando los coeficientes de X^2, obtenemos la igualdad:

$$-\lambda^2 = -p_1 - q_1 - r_1,$$

y despejando, obtenemos:

$$r_1 = \lambda^2 - p_1 - q_1.$$

Usando la ecuación de la recta, obtenemos:

$$r_2 = \lambda r_1 + (p_2 - \lambda p_1).$$

Como el punto R es la simetría respecto del eje X de (r_1, r_2), tenemos:

$$R = (r_1, -r_2) = (\lambda^2 - p_1 - q_1, -\lambda r_1 - (p_2 - \lambda p_1)).$$

□

2.2.1. Asociatividad en $(E(K), +)$

Esta sección tiene como objetivo probar la asociatividad de la suma definida anteriormente para puntos de curvas elípticas. Con esto quedaría concluida la prueba de la ley de grupo (teorema 2.2.1).
La principal referencia de esta sección ha sido [5], capítulo 5.
La primera forma de probar la propiedad asociativa es utilizando las fórmulas explícitas de la suma, que hemos dado en el teorema 2.2.2. Sin embargo, esta prueba es tediosa.
Por ello, vamos a probar la asociatividad de manera geométrica; por este motivo hemos definido las curvas planas con multiplicidad como lo hemos hecho en la definición 1.2.8. Necesitamos una serie de resultados para poder continuar. Con el objetivo de hacer la prueba más legible, introducimos la siguiente notación:

Definición 2.2.2. *Sea C una curva cúbica (de grado 3) irreducible, no necesariamente regular, cualquiera, y P, Q dos puntos de la misma. Sea $L = PQ$, la recta que pasa por P y Q. Además, L tiene que pasar necesariamente por otro punto de la curva, que llamaremos R, ya que C tiene grado 3.*
Denotamos $L \bullet C = P \oplus Q \oplus R$, a los tres puntos de la intersección (la intersección de la recta L con la curva C da tres puntos, P, Q y R, que son los puntos de intersección); esto es una suma formal de puntos.
La ventaja que obtenemos, es que contamos los puntos con sus multiplicidades, lo que significa que si algún punto se repite (por ejemplo, si hay una tangencia), se cuenta más de una vez.
Definimos $\phi : C \times C \to C$, como $\phi(P, Q) = R$.

Nota. *Si C es una curva elíptica en forma de Weierstrass simplificada, se puede comprobar fácilmente que la suma de dos de sus puntos P y Q, como dimos en la definición 2.2.1, cumple que $P + Q = \phi(O, \phi(P, Q))$. Omitimos los detalles por brevedad.*

Proposición 2.2.2. *(**Versión del Teorema de Cayley-Bacharach**)*
Sea C una curva elíptica irreducible, y C', C'' dos curvas cúbicas. Si tenemos que $C' \bullet C = \bigoplus_{i=1}^{9} P_i$, donde P_i son puntos simples de la curva, y también $C'' \bullet C = \bigoplus_{i=1}^{8} P_i \bigoplus Q$, donde Q es un punto no singular de la curva C. Entonces, el noveno punto de la intersección, P_9, coincide con Q.

Demostración. La prueba de esta proposición requiere una teoría que se sale de los objetivos del trabajo. Se puede encontrar la prueba detallada en [5], proposición 3 del capítulo 5.

□

Utilizando esto, podemos demostrar la propiedad asociativa [4]:

Demostración. Utilizando la definición anterior, vamos a demostrar la propiedad asociativa de la suma de puntos: sean P, Q, R tres puntos de $E(K)$, veamos que $(P+Q)+R = P+(Q+R)$. Para ello, calcularemos los puntos de ambos lados de la igualdad y usaremos las intersecciones de las rectas definidas a continuación.

Sea $L_1 = PQ$; escribimos entonces $L_1 \bullet C = P \oplus Q \oplus S'$, donde $S' \in C$ es el tercer punto de intersección de la recta L_1 con C.
Sea $L_2 = OS'$; escribimos entonces $L_2 \bullet C = O \oplus S' \oplus S$, donde $S \in C$ es el tercer punto de intersección de la recta L_2 con C.
Así, concluimos que $S = P + Q$.
Sea $L_3 = SR$; escribimos $L_3 \bullet C = S \oplus R \oplus T'$, con $T' \in C$ el tercer punto de la intersección de L_3 con C.
Sea $L_4 = OT'$; escribimos $L_4 \bullet C = O \oplus T' \oplus T$, con $T \in C$ el tercer punto de la intersección de L_4 con C.
Así, obtenemos que $T = (P + Q) + R$.
Ahora, consideremos lo siguiente:
Sea $M_1 = QR$; escribimos $M_1 \bullet C = Q \oplus R \oplus U'$, con $U' \in C$ el tercer punto de la intersección de M_1 con C.
Sea $M_2 = U'O$; escribimos $M_2 \bullet C = O \oplus U' \oplus U$, con $U \in C$ el tercer punto de la intersección de M_2 con C.
Tenemos $U = Q + R$.
Sea $M_3 = PU$; escribimos $M_3 \bullet C = P \oplus U \oplus V'$, con $V' \in C$ el tercer punto de la intersección de M_3 con C.
Sea $M_4 = OV'$; escribimos $M_4 \bullet C = O \oplus V' \oplus V$, con $V \in C$ el tercer punto de la intersección de M_4 con C.
Así, obtenemos que $V = P + (Q + R)$.
Vamos a demostrar que $T' = V'$, ya que entonces las rectas L_4 y M_4 coinciden y, por lo tanto, $T = V$, que es exactamente lo que necesitamos.
Si $C_1 = L_1 \cdot M_2 \cdot L_3$, entonces $C' = C_1 \bullet C = P \oplus Q \oplus S' \oplus O \oplus U' \oplus U \oplus S \oplus R \oplus T'$.
Por otro lado, si $C_2 = M_1 \cdot L_2 \cdot M_3$, entonces $C'' = C_2 \bullet C = Q \oplus R \oplus U' \oplus O \oplus S' \oplus S \oplus P \oplus U \oplus V'$.
Dado que C' y C'' son dos cúbicas que coinciden en 8 puntos: P, Q, S', O, U', U, S, R y T, por la proposición anterior, podemos concluir que coinciden en un noveno punto, es decir, $T' = V'$.
Por lo tanto, $(P + Q) + R = P + (Q + R)$. □

Existen pruebas de la asociatividad sin recurrir a que el cuerpo sobre el que se define la curva sea algebraicamente cerrado, utilizando por ejemplo la versión proyectiva de una curva elíptica. Se puede ver en [2]. Esto nos permite generalizar la ley de grupo para curvas elípticas sobre cualquier cuerpo.

2.3. Curvas elípticas sobre cuerpos finitos

Con el objetivo de utilizar curvas elípticas en el ámbito de la criptografía, es necesario que el grupo que se utilice sea finito. Con este objetivo, se presentan las curvas elípticas sobre un cuerpo finito, $E(\mathbb{F}_q)$ (con $q = p^n$, p primo, $n \in \mathbb{N}$), que forman un grupo conmutativo y finito. En adelante, consideraremos solo el caso $E(\mathbb{F}_p)$, con p primo, por simplicidad en las cuentas. Sin embargo, todos los resultados se cumplirán igualmente para $E(\mathbb{F}_q)$, donde $q = p^n$ para p primo y $n \in \mathbb{N}$ cualquiera.
En criptografía, es habitual que el grupo finito tenga un cardinal alto, por lo que para lo que resta del trabajo, $p \neq 2, 3$.
Si tenemos una curva elíptica $E(\mathbb{F}_p)$, podemos tomar congruencias módulo un número primo (en nuestro caso va a ser distinto de 2 y 3) que no divida al discriminante de la curva, y de esta manera, obtendremos la ecuación de una curva elíptica definida sobre un cuerpo finito.

Definición 2.3.1. *Definimos una* ***curva elíptica sobre un cuerpo finito*** $\mathbb{F}_p$ *con la ecuación* $Y^2 = X^3 + AX + B$*, con la condición* $4A^3 + 27B^2 \neq 0$*. Los puntos de la curva,*$E(\mathbb{F}_p)$*, serán las soluciones* (X, Y) *en* $\mathbb{F}_p$ *que satisfacen esa ecuación, junto con el punto del infinito,* O*.*

Nota. *Sobre las curvas elípticas sobre cuerpos finitos aplican las propiedades vistas en el capítulo anterior. En particular,son un grupo con la suma definida en la definición 2.2.1.*

Ejemplo 2.3.1. *Sea* E *la curva definida por la ecuación* $Y^2 = X^3 + 2X + 1$ *sobre* $\mathbb{F}_{13}$*. Calculemos los puntos de esta curva,* $E(\mathbb{F}_{13})$*, explícitamente. Para ello, vamos a sustituir los valores* $X = 0, 1, .., 12$*, y comprobemos cuando* Y *es un valor exacto módulo 13.*
Para $X = 0$*,* $Y^2 = 1 \mod 13$*, y por tanto,* $Y = 1 \mod 13$ *o* $Y = -1 \mod 13 \equiv 12 \mod 13$ *.*
Para $X = 1$*,* $Y^2 = 4 \mod 13$*, y por tanto,* $Y = 2 \mod 13$ $Y = -2 \mod 13 \equiv$*.*
Para $X = 2$*,* $Y^2 = 0 \mod 13$*, y no tiene solución .*
Haciendo esto para todos los posibles valores de X*, obtenemos que la curva consta de los puntos:*

$$E(\mathbb{F}13) = \{O, (0,1), (0,12), (1,2), (1,11), (0,2), (8,3), (8,10)\}.$$

$E(\mathbb{F}_{13})$ *tiene orden 8.*
Podemos ver la curva definida gráficamente:

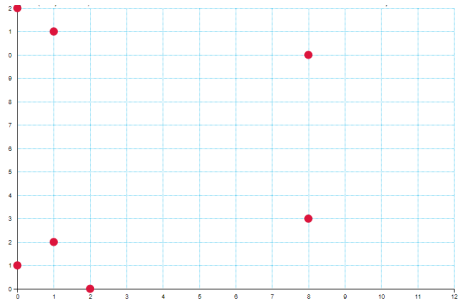

Figura 2.4: Curva elíptica $y^2 = x^3 + 2x + 1$

Por último, vamos a sumar los puntos $(0,1)$ *y* $(1,2)$*, y el punto* $(1,2)$ *consigo mismo.*

1. $(0,1) + (1,2)$*:*
   $\lambda = \frac{2-1}{1-0} = 1$*.Por tanto,* $x_3 = 1^2 - 0 - 1 = 0$ *y* $y_3 = 1(0-0) - 1 = -1 = 12$ mód 13.
   Entonces, $(0,1) + (1,2) = (0,12)$.

2. $(1,2) + (1,2)$*:*
   $\lambda = \frac{3,1^2+2}{2,2} = 5/4 = 11 \mod 13$*. Por tanto,* $x_3 = 11^2 - 1 - 1 = 119 = 2$ mód 13*, y* $y_3 = 11(1-2) - 2 = -13 = 0 \mod 13$.
   Entonces, $(1,2) + (1,2) = (2,0)$.

2.3.1. Número de puntos de una curva

Es interesante intentar acotar lo más finamente posible el cardinal de una curva elíptica sobre un cuerpo finito.
Para esta sección se ha usado como referencia [14], [16] y [3].

Proposición 2.3.1. *El número máximo de puntos que puede tener una curva elíptica sobre el cuerpo finito* $\mathbb{F}_p$ *es* $2p + 1$.

Demostración. Sea E una curva elíptica sobre el cuerpo finito $\mathbb{F}_p$. Entonces, para los puntos de $E(\mathbb{F}_p) = (X_i, Y_i)$, los valores que puede tomar X son, a lo sumo, el cardinal del cuerpo, es decir $|\mathbb{F}_p| = p$. Por otro lado, al despejar Y de la ecuación en forma de Weierstrass de la curva, $Y = \pm\sqrt{X^3 + 3AX + B}$, por lo que tiene a lo sumo dos valores para cada X. Por último, hay que tener en cuenta el punto del infinito. Con todo esto, obtenemos que, como mucho, $|E(\mathbb{F}_p)| = 2p + 1$. □

Esta cota, sin embargo, es muy superior al tamaño real, y queremos una mejor estimación.

Teorema 2.3.1. *(**Teorema de Hasse**) Si E es una curva elíptica definida sobre el cuerpo finito $\mathbb{F}_p$, entonces el número de puntos, $E(\mathbb{F}_p)$, satisface $|E(\mathbb{F}_p)| = p + 1 - t_p$, donde $0 \leq |t_p| \leq 2\sqrt{p}$.*

Para poder demostrar este teorema, necesitamos hablar del endomorfismo de Frobenius. Antes, recordaremos la definición de endomorfismo separable.

Definición 2.3.2. *Un **endomorfismo de** E es un homomorfismo de grupos $\alpha : E(\overline{K}) \longrightarrow E(\overline{K})$, tal que $\alpha(x, y) = (R_1(x, y), R_2(x, y))$, donde $R_1(x, y), R_2(x, y)$ son funciones racionales, y $\overline{K}$ representa la clausura algebraica del cuerpo K.*

Nota. *La relación algebraica entre x e y en la curva permite eliminar la dependencia explícita de y de la primera función racional, y expresarla en términos de x, ya que la segunda variable quedará definida por la condición impuesta por la curva.*
Así, podemos reescribir un endomorfismo definido sobre una curva elíptica (en forma de Weierstrass) como $\alpha(x, y) = (r_1(x), r_2(x)y)$, con r_1, r_2 funciones racionales en x. Podemos encontrar el proceso detallado en [18], en la sección 2.9.

Definición 2.3.3. *Un endomorfismo α no trivial, dado por $\alpha(x, y) = (r_1(x), r_2(x)y)$ es **separable** si $r_1'(x) \neq 0$.*

Definición 2.3.4. *Sea $\alpha \neq 0$, un endomorfismo sobre una curva elíptica E, de la forma $\alpha(x, y) = (r_1(x), r_2(x)y)$, con r_1, r_2 funciones racionales. Consideramos $r_1(x) = \frac{p(x)}{q(x)}$ en su forma más simplificada. Entonces se define su **grado** como $\deg \alpha$, como el máximo grado entre $p(x)$ y $q(x)$, es decir, $\deg(\alpha) = \max\{\deg(p(x)), \deg(q(x))\}$.*
Si el endomorfismo es nulo (es decir, el que lleva todo punto al elemento neutro de la curva, O), su grado será 0.

Definición 2.3.5. *Sea $\mathbb{F}_p$ un cuerpo finito. Definimos el **endomorfismo de Frobenius**: $\phi : E(\overline{\mathbb{F}_p}) \longrightarrow E(\overline{\mathbb{F}_p})$ como la aplicación dada por $\phi(x, y) = (x^p, y^p)$ y $\phi(O) = O$. El grado de este endomorfismo es p.*

Nota. *La aplicación dada es un endomorfismo, ya que ϕ está bien definida, y lleva puntos de la curva en puntos de la curva: sustituyendo el valor del punto (x^p, y^p) en la ecucación simplificada de la curva de Weierstrass, esta se sigue satisfaciendo.*
Además, es un homomorfismo de grupos, ya que preserva la suma (simplemente podríamos operar, utilizando la propiedad de que en cuerpos de característica p, $(x + y)^p = (x^p + y^p)$, con $x, y \in E(\overline{\mathbb{F}_p})$).

Nota. *El endomorfismo de Frobenius no es separable, ya que si $r_1 = x^p$, entonces $r_1'(x) = p \cdot x^{p-1} = 0$ en un cuerpo de característica p. Además, al estar definido por funciones racionales en su forma más simple, y de la forma $r_1(x) = \frac{x^p}{1}$, el grado del mayor de las dos es p.*

A continuación daremos unas proposiciones cuyas pruebas podemos encontrar en : [19] y [18].

Proposición 2.3.2. *Sea α un endomorfismo separable no nulo, sobre una curva elíptica sobre un cuerpo cualquiera K. Entonces, $deg(\alpha) = |Ker(\alpha)|$.*
Si el endomorfismo no es separable, entonces, $deg(\alpha) > |Ker(\alpha)|$.

La prueba detallada se encuentra en [18], en la proposición 2.21 de la sección 2.9.

Proposición 2.3.3. *Dado $P = (x, y) \in E(\overline{\mathbb{F}_p})$, tenemos que $P \in E(\mathbb{F}_p)$ si y sólo si $\phi(P) = P$.*

Demostración. $\Rightarrow$) Si $P = (x, y) \in E(\mathbb{F}_p)$, entonces cumple la ecuación de Weierstrass, $y^2 = x^3 + Ax + B$. Elevando las coordenadas, obtenemos $(y^p)^2 = (x^p)^3 + A(x^p) + B$. Como $x^p = x$ e $y^p = y$ porque $x, y \in \mathbb{F}_p$, entonces $\phi(P) = P$.
$\Leftarrow$) Si $\phi(P) = (x^p, y^p) = (x, y) = P$, entonces $x^p = x$ e $y^p = y$, por lo que $x, y \in \mathbb{F}_p$ y $P \in E(\mathbb{F}_p)$. □

Proposición 2.3.4. *Sea α el endomorfismo de Frobenius sobre la curva elíptica $E(\overline{\mathbb{F}_p})$, y sean $r, s \in \mathbb{Z}$ no nulos simultáneamente. Entonces, el endomorfismo $r\alpha + s$ es separable si y sólo si p no divide a s.*

Nota. *En la proposición anterior, llamamos $r\alpha + s$ al homomorfismo $r\alpha + s1$, donde 1 es el endomorfismo identidad, y donde, para un endomorfismo ϕ, la notación $r\phi$ quiere decir el endomorfismo que lleva el punto P en $\phi(P) + \cdots + \phi(P)$, sumado consigo mismo r veces.*

La prueba detallada, que no concretamos por requerir herramientas de endomorfismos que salen del marco del trabajo, se encuentra en [18], en la proposición 2.29 de la sección 2.9.

Proposición 2.3.5. *Sea α el endomorfismo de Frobenius sobre la curva elíptica $E(\overline{\mathbb{F}_p})$. Entonces, $Ker(\alpha - 1) \cap E(\overline{\mathbb{F}_p}) = E(\mathbb{F}_p)$, siendo 1 la identidad.*

Demostración. Es consecuencia inmediata de la proposición 2.3.3. □

Proposición 2.3.6. *Sea p un primo y $r, s \in \mathbb{Z}$ tales que s y p sean primos entre sí. Si $a = p + 1 - \deg(\phi - 1)$, siendo ϕ el endomorfismo de Frobenius sobre $E(\overline{\mathbb{F}_p})$, entonces, $\deg(r\phi - s) = r^2p + s^2 - rsa$.*

Demostración. La idea de la demostración es que $\deg(r\phi - s) = r^2 \deg(\phi) + s^2 \deg(-1) + rs(deg(\phi - 1) - \deg(\phi) - \deg(-1))$. Podemos encontrar una demostración detallada de los cálculos en la proposición 3.16 de [18]. □

Estamos en condiciones de demostrar el Teorema de Hasse: 2.3.1

Demostración. Consideramos el endomorfismo de Frobenius definido sobre $E(\overline{\mathbb{F}_p})$. Como consecuencia de las proposiciones 2.3.5 y 2.3.2, por ser $\phi - 1$ un endomorfismo separable (tomando $r = 1, s = -1$), entonces $|\deg(\phi - 1)| = |Ker(\phi - 1)| = |E(\mathbb{F}_p)|$. Por la proposición 2.3.6, $\deg(\phi - 1) = p + 1 - a$, despejamos y $a = p + 1 - \deg(\phi - 1)$.

Falta demostrar que $|a| \leq 2\sqrt{p}$: la proposición 2.3.6 implica de manera inmediata que $\deg(r\phi - s) \geq 0$, es decir, $p(\frac{r}{s})^2 + 1 - a(\frac{r}{s}) \geq 0$. El conjunto de números racionales es denso en los reales, por lo que la inecuación se mantiene para $x \in \mathbb{R}$. Por tanto, el discriminante de esta debe ser menor o igual que 0, por lo que $a^2 - 4p \leq 0$. Despejando, obtenemos que $|a| \leq 2\sqrt{p}$.
Con esto, recordando que teníamos $|E(\mathbb{F}_p)| = \deg(\phi - 1)$, obtenemos que $|E(\mathbb{F}_p)| = p + 1 - a$ y $|a| \leq 2\sqrt{p}$. Terminamos definiendo $t_p = a$. □

Este resultado es importante, no sólo porque proporciona una cota más óptima del número de puntos de una curva elíptica definida sobre un cuerpo finito, sino porque además podemos afirmar que la curva tendrá algún punto. De hecho, el número de puntos de $E(\mathbb{F}_p) \approx p$, ya que el término $t_p \ll p$ (remarcamos que no es una igualdad).
Usando el número de puntos de una curva, podemos establecer una distinción:

Definición 2.3.6. *Sea $E(\mathbb{F}_p)$ una curva elíptica.*

- *Una curva elíptica se denomina **supersingular** si $|E(\mathbb{F}_p)| \equiv 1 \mod p$.*
- *Una curva elíptica se denomina **anómala** si $|E(\mathbb{F}_p)| = p$.*

En el ámbito criptográfico, en numerosas ocasiones necesitamos saber exactamente el orden del grupo sobre el que trabajamos. Comentaremos sin entrar al detalle algunos de los métodos para calcular el número de puntos de una curva elíptica sobre un cuerpo finito:

1. Algoritmo de cálculo ingenuo: calcular los cuadrados en $\mathbb{F}_p$ y compararlos con la ecuación $Y = \pm\sqrt{X^3 + 3AX + B}$. Esto es extremadamente ineficiente, ya que refiere un coste computacional de p^2 operaciones, ya que hay que probar todos los elementos del cuerpo.

2. Si $E(\mathbb{F}_p)$ es cíclico, y P un elemento, calcular el subgrupo generado por P. El orden del grupo, n, es el primer entero positivo tal que $nP = O$. Este método tampoco es eficiente computacionalmente si p es un primo grande, ya que el coste computacional podría llegar a ser $O(\log p)$, si el tamaño del grupo se aproxima demasiado a p.

3. Algoritmo de Schoof. Este algoritmo fue publicado en 1985 por René Schoof. La idea del mismo, es utilizar la igualdad dada por el teorema de Hasse, hallando explícitamente t_p.Podemos encontrar todo el detalle y la demostración en [17].
   La idea del algoritmo es ir reduciendo módulo p_i, con varios primos más pequeños, el problema de encontrar t_p, y con los resultados obtenidos, utilizar el teorema chino de los restos para combinar los resultados y a la postre, obtener el orden del grupo.
   Este algoritmo es el más eficiente de los tres de manera general, teniendo un coste computacional de aproximadamente $O((\log p)^4)$, siendo principalmente el coste de calcular los módulos respecto a los primos pequeños.

2.3.2. Multiplicación de puntos

Para el desarrollo de criptosistemas, necesitamos saber cómo calcular kP, siendo P un punto de una curva elíptica definida sobre un cuerpo finito, $\mathbb{F}_p$, y k un entero.

Definición 2.3.7. *Sea $n \in \mathbb{Z}$ y $P \in E(K)$, para un cuerpo K. Definimos la operación* ***multiplicación por un entero*** *en una curva elíptica por la siguiente aplicación:*

$$n\bullet : E(K) \longrightarrow E(K)$$

definida como sigue:

1. *Si $n = 0$, $0 \bullet P = O$.*
2. *Si $n > 0$, $n \bullet P = P + \cdots + P$ (n veces la suma del punto).*
3. *Si $n < 0$, $n \bullet P = -n(-P)$.*

Nota. *En adelante, omitiremos la notación $\bullet$ como hacemos en el producto habitual.*

Definición 2.3.8. *Dado un punto $P \in E(K)$, donde K es un cuerpo, diremos que P tiene orden finito si existe un entero positivo n tal que $nP = O$. El menor de los enteros positivos que lo cumple se denomina* ***orden de*** *P.*

A continuación veremos dos algoritmos para realizar la multiplicación de un punto por un entero, dentro de las curvas elípticas. Para ello, consideraremos que $\#E(\mathbb{F}_p) = n \cdot \epsilon$, con n primo y ϵ pequeño. Además, P, Q tienen orden n. En los siguientes algoritmos consideraremos $k \in [1, n-1]$ un entero aleatorio, y su representación binaria, $(k_{t-1}, ..., k_1, k_0)$.

- Algoritmo de punto desconocido.
  Este método se basa en duplicar los puntos de manera eficiente. Este cálculo tiene una complejidad de $O(\log(k))$, debido a que hay $O(\log k)$ bits, el número total de iteraciones (sumas) es proporcional a $O(\log k)$.
  Hay dos versiones, método binario de derecha a izquierda, y de izquierda a derecha.

Algorithm 1 Cálculo en binario de derecha a izquierda, $kP = Q$

Input: k, P.
Output $kP = Q$
$Q \leftarrow \infty$
for i desde 0 hasta $t-1$ **do** s
  if $k_i = 1$ **then**
    $Q \leftarrow Q + P$
  end if
  $P \leftarrow 2P$
end for
Return Q.

Algorithm 2 Cálculo en binario de izquierda a derecha, $kP = Q$

Input: k, P.
Output $kP = Q$
$Q \leftarrow \infty$
for i desde $t-1$ hasta 0 **do**
  $Q \leftarrow 2Q$
  if $k_i = 1$ **then**
    $Q \leftarrow Q + P$
  end if
end for
Return Q.

- Forma no adyacente (NAF).
  Este algoritmos se basa en que la resta de puntos de una curva elíptica es igual de eficiente que la suma (si $P = (x, y)$ un punto de $E(\mathbb{F}_q)$, entonces $-P = (x, -y)$, ya que suponemos la característica distinta de 2). Por tanto, queremos usar una representación de k de la forma, $k = \sum_{i=0}^{l-1} k_i 2^i$, con $k_i = 0, -1, 1$. Una representación de este tipo usada para calcular la multiplicación de punto por escalar es el NAF (non adjacent form).

  Definición 2.3.9. *La **forma no adyacente (NAF)** de un entero positivo k es una expresión de la forma $NAF(k) = \sum_{i=0}^{l-1} k_i 2^i$, con $k_i = \{0, -1, 1\}$, con k_{l-1} no nulo, y sin dos k_i consecutivos no nulos. La longitud del NAF es l.*

  En primer lugar, veamos un algoritmo para calcular el NAF de un número natural.

Algorithm 3 Cálculo de NAF(k) con k entero positivo

Input: k.
Output $NAF(k)$
$i \leftarrow 0$
while k ≤ 1 **do**
  if k es impar **then**
    $k_i \leftarrow 2 - (k \mod 4)$, $k \leftarrow k - k_i$
  else
    $k_i \leftarrow 0$
  end if
  $k \leftarrow k/2$, $i \leftarrow i + 1$
end while
Return $(k_{i-1}, k_{i-2}, \cdots, k_0)$.

Nuestro objetivo es usar el NAF de k para computar la multiplicación de un punto de la curva por un escalar. Veamos una serie de propiedades de

la forma no adyacente de k.

Proposición 2.3.7. *1. Cada k tiene un único NAF(k).*

2. *La longitud de NAF(k) es al menos una mayor que la longitud de la representación binaria de k.*
3. *NAF(k) tiene menos dígitos nulos que cualquier otra representación de k.*

Demostración. 1. Unicidad: razonemos por reducción al absurdo. Supongamos que existen dos representaciones distintas en forma de NAF para $k \in \mathbb{Z}$: $NAF_1(K) = \sum_{i=0}^{l-1} k_i 2^i$ y $NAF_2(K) = \sum_{i=0}^{j-1} k'_i 2^i$ que cumplen las propiedades de la definición 2.3.9, y tales que $NAF_1(K) \neq NAF_2(K)$. Como el procedimiento para obtener el NAF es recursivo, y siempre se calcula de la misma manera, como hemos visto en el algoritmo anterior. Para cada k_i, el número siguiente se ajusta para que no haya coeficientes consecutivos no nulos. Esto garantiza que el proceso de descomposición es único.

2. Longitud: La representación binaria de un número k tiene una longitud de n, con $2^{n-1} \leq k < 2^n$.
En la forma NAF de k, cada k_i puede ser $0, 1$ o -1, pero no puede haber dos coeficientes consecutivos no nulos. Tenemos en cuenta que Solo los coeficientes 1 y -1 son los que determinan la longitud de la representación NAF, ya que son los que suman potencias de 2 a la expansión. Para garantizar que se cumple la propiedad, tras la presencia de algún $k_i = \pm 1$, que se asignaría al bit en binario, necesitaríamos añadir un bit extra de longitud para garantizar que los coeficientes consecutivos no sean no nulos. Esto implica que la longitud del NAF será al menos una mayor que la binaria.

3. Menos dígitos nulos: esto es consecuencia directa de la restricción de que no hay dos coeficientes consecutivos no nulos (alno permitir coeficientes consecutivos no nulos en la representación NAF obliga a que se utilicen las representaciones de 1 y -1 de manera eficiente. Al utilizar 1 y -1, se reduce la necesidad de colocar más unos consecutivos, lo que reduce la cantidad de ceros).

□

Veamos un ejemplo del cálculo del NAF(k) y su diferencia con la representación binaria:

Ejemplo 2.3.2. *Tomamos* $k = 7$.
Representación binaria de k: $7 = 1{,}2^2 + 1{,}2^1 + 1{,}2^0 \rightarrow 7 = 111$.
Representación NAF(k): $7 = 1{,}2^3 + 0{,}2^2 + 0{,}2^1 - 1{,}2^0 \rightarrow 7 = 100 - 1$.

Algorithm 4 Multiplicación de puntos mediante NAF

```
Input: k, P.
Output kP
Calcular NAF(k).
Q ← ∞.
for i desde l − 1 hasta 0 do
    Q ← 2Q.
    if k_i = 1 then
        Q ← Q + P
    else
        if k_i = −1 then
            Q ← Q − P
        end if
    end if
end for
Return Q.
```

2.3.3. Puntos de Torsión y emparejamiento de Weil

En esta última sección de la teoría matemática de las curvas elípticas, comentaremos el concepto de puntos de torsión de una curva y del emparejamiento de Weil. Utilizamos como principales referencias [18] y [14].

Definición 2.3.10. *Sea K un cuerpo (no necesariamente finito) y consideramos $E(K)$ una curva elíptica. Sea n un entero positivo. El conjunto de puntos:*

$$E[n] = \{P \in E(K)/nP = O\}$$

se denomina el conjunto de ***puntos de n-torsión*** *de la curva E.*

Teorema 2.3.2. *En las condiciones de la definición, $E[n]$ es un subgrupo de E.*

Demostración. Sean $P, Q \in E[n]$. En primer lugar, el elemento neutro, O, pertenece a $E[n]$ por definición.
Veamos ahora que es cerrado para la suma y para los inversos, sabiendo que $nP = nQ = O$.
Tenemos que $n(P+Q) = nP + nQ = O + O = O$, por lo que $P + Q \in E[n]$.
Sabiendo que $P + (-P) = O$ y multiplicando por el natural n, obtenemos $O = n(P+(-P)) = nP+n(-P)$, por lo que $n(-P) = O$. Luego $-P \in E[n]$. □

Definición 2.3.11. *Dada una curva $E(K)$, definimos el* ***subgrupo de torsión de E*** *como:*

$$E_t = \bigcup_{n=1}^{\infty} E[n].$$

Teorema 2.3.3. *Sea $E(\mathbb{F}_p)$ y $n \in \mathbb{N}$ tal que p no divide a n. Entonces,*

$$E(\mathbb{F}_{p^{jk}})[n] \cong \mathbb{Z}_n \times \mathbb{Z}_n,$$

para algún $k \in \mathbb{N}$ y para todo $j \in \mathbb{N}$.

Se puede encontrar una prueba en [14], pero no se entra en detalle ya que sobrepasa el objetivo del trabajo.

Definición 2.3.12. *Al entero positivo más pequeño k que cumple que*

$$E(\mathbb{F}_{p^k})[n] \cong \mathbb{Z}_n \times \mathbb{Z}_n$$

*le llamaremos **grado de extensión** de E respecto de n.*

Por último, se hablará del emparejamiento de Weil, que se usa como base de un ataque criptográfico específico para algunos tipos de curvas elípticas. Necesitamos hablar brevemente del grupo de divisores de una curva.

Definición 2.3.13.

- *Sea E una curva elíptica. Se define un **divisor** $D = \sum_{P \in E} n_p[P]$, donde $n_P \in \mathbb{Z}\backslash\{0\}$ para un conjunto finito de puntos $P \in E$, e indica la multiplicidad con la que cada punto P aparece en D. $[P]$ es un símbolo para representar al punto P de la curva en el divisor.*

  Nota. *Se utiliza la notación $\sum_{P \in E} n_P[P]$ ya que $\sum_{P \in E} n_P \cdot P$ denota la suma de puntos usual en la curva elíptica.*

- *Denominamos $Div(E)$ al **grupo de divisores de la curva** E, con la operación $D + D' = \sum_{P \in E}(n_P + n'_P)[P]$, para D, D' divisores.*
- *El **soporte de un divisor** D es el conjunto de puntos con coeficiente n_P no nulo.*
- *Se define **grado de un divisor** D como $\deg D = \sum_{P \in E} n_P$.*
- *Dada una función racional definida sobre E, $f(x, y) = \frac{g(x,y)}{h(x,y)}$, con $g, h \in K[x, y]$, y dado $P \in E$, P es un **cero** de f si $g(P) = 0$ y $h(P) \neq 0$. Si $h(P) = 0$ y $g(P) \neq 0$, se dice que P es un **polo** de f.*
- *Llamamos **divisor asociado a una función racional** f a $div(f) = \sum_{P \in E} ord_P(f)[P]$, donde $ord_P(f) = n$ si P es un cero con multiplicidad n de f y $ord_P(f) = -n$ si P es un polo de multiplicidad n.*
- *Diremos que dos divisores, D, D' son **equivalentes** si existe una función racional f tal que $D = D' + div(f)$.*

Teorema 2.3.4. *Si E es una curva elíptica definida sobre un cuerpo K, y D un divisor de grado 0 de la curva de la forma $D = \sum_P n_P[P]$, entonces existe una función racional f tal que $D = div(f)$ si y sólo si $\sum_P n_P \cdot P = O$, (donde $\sum_P n_P \cdot P$ se refiere a la suma de puntos en la curva elíptica).*

Encontramos la prueba en [18], en el teorema 11.2.

Definición 2.3.14. *Sea $E(\mathbb{F}_p)$ una curva elíptica, y sean $P, Q \in E[n]$, con $n \in \mathbb{N}$, tal que p no divide n. En estas condiciones, definimos el* ***emparejamiento de Weil*** *como una aplicación:*

$$e_n : E[n] \times E[n] \longrightarrow \{x \in \overline{\mathbb{F}}_p \mid x^n = 1\},$$

que cumple las siguientes propiedades:

1. *e_n es una aplicación bilineal. Es decir, $\forall P, Q_1, Q_2 \in E[n]$, se tiene*

$$e_n(P, Q_1 + Q_2) = e_n(P, Q_1)e_n(P, Q_2),$$

$$e_n(Q_1 + Q_2, P) = e_n(Q_1, P)E_n(Q_2, P).$$

2. *e_n es una aplicación no degenerada; es decir, si existe $P \in E[n]$ tal que $\forall Q \in E[n]$ se cumple que $e_n(P, Q) = 1$, entonces $P = O$.*
3. *Si $P \in E[n]$, entonces $e_n(P, P) = 1$.*
4. *Para todo σ automorfismo de $\overline{K}$ que fija a los coeficientes de E, $e_n(\sigma(P), \sigma(Q)) = \sigma(e_n(P, Q))$.*
5. *Para todo endomorfismo separable de E, σ, $e_n(\sigma(P), \sigma(Q)) = e_n(P, Q)^{\deg(\sigma)}$.*

Explícitamente, se puede definir un emparejamiento de Weil de la siguiente manera:
Sean $P, Q \in E[n]$, y D, D' divisores de grado cero con soportes disjuntos y tales que D es equivalente a $P - O$ y D' es equivalente a $Q - O$. Sean f_D y $f_{D'}$ dos funciones racionales sobre E tales que $div(f_D) = nD$ y $div(f_{D'}) = nD'$ (por el teorema 2.3.4 sabemos que exitirán).
Podemos definir el emparejamiento de Weil como:

$$e_n(P, Q) = \frac{f_D(D')}{f_{D'}(D)}.$$

Se puede encontrar más detalle en [18] y [12], en el capítulo 5.

Capítulo 3

Criptografía basada en el problema del logaritmo discreto

En este capítulo, exploraremos la aplicación de los conceptos desarrollados en el capítulo anterior al campo de la criptografía. Comenzaremos con una introducción a los fundamentos generales de la criptografía, definiendo los distintos modelos criptográficos utilizados en la actualidad. A continuación, profundizaremos en cómo estos modelos se implementan sobre el grupo de curvas elípticas, una componente clave en la criptografía moderna.
Las principales referencias del capítulo son: [15], [7] y [18].

3.1. Conceptos generales

En esta sección, presentaremos los conceptos fundamentales de la criptografía y profundizaremos en aquellos aspectos que serán clave para los modelos criptográficos que analizaremos a lo largo del capítulo.

Definición 3.1.1. *La **criptografía** es la ciencia encargada de garantizar la seguridad en la comunicación de mensajes mediante procesos de codificación (cifrado) y decodificación (descifrado). El cifrado de un mensaje se realiza utilizando un **algoritmo criptográfico**, cuyo funcionamiento es conocido públicamente. Sin embargo, para asegurar la confidencialidad del mensaje, se requiere una herramienta adicional para el descifrado, la cual se denomina **clave**.*

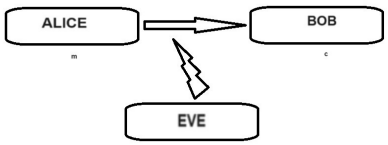

Figura 3.1: Esquema criptografía

Como se puede observar en la figura 3.1, en el proceso de transmisión segura de un mensaje intervienen varias personas:

Definición 3.1.2.

- ***Emisor*** *(Alice): Es la persona que posee el mensaje en su forma original (texto plano) y lo cifra utilizando un algoritmo criptográfico.*
- ***Receptor*** *(Bob): Es quien recibe el mensaje cifrado, que puede estar sujeto a interferencias, y debe descifrarlo para obtener el mensaje original.*
- ***Analista*** *(Eve): Comúnmente conocida como "hacker", esta persona puede intentar interceptar el mensaje o interferir en su transmisión con el fin de modificarlo. Su objetivo es conseguir el contenido del mensaje original enviado por Alice a partir del mensaje cifrado. La ciencia que estudia estas amenazas es el* ***criptoanálisis****.*

Definición 3.1.3. *Existen dos tipos principales de algoritmos criptográficos:*

- *Algoritmos de* ***clave privada o simétricos****: Tanto el emisor como el receptor utilizan la misma clave para cifrar y descifrar el mensaje. Solo el emisor y el receptor pueden conocerla.*
- *Algoritmo de* ***clave pública o asimétricos****: La clave para cifrar el mensaje es diferente de la clave para descifrarlo. Este es el tipo de algoritmo sobre el que nos centraremos en este trabajo.*
  Los algoritmos de clave pública surgen de la necesidad de evitar el intercambio previo de una clave secreta, que puede ser inviable o inseguro a través de canales públicos.
  Los elementos fundamentales de un sistema de criptografía asimétrica son: $K = (k_{pub}, k_{priv})$, donde k_{pub} es la ***clave pública*** *y k_{priv} es la* ***clave privada****; M es el* ***conjunto de mensajes en texto plano****, y C es el* ***conjunto de mensajes cifrados****. Además, se utilizan dos funciones: una para* ***cifrado*** *y otra para* ***descifrado****. Estas funciones, para las claves pública y privada, se expresan como $e_{k_{pub}} : M \to C$ y $d_{k_{priv}} : C \to M$. Es fundamental que estas funciones sean inversas una de la otra: $\forall m \in M, d_{k_{priv}}(e_{k_{pub}}(m)) = m$.*

Definición 3.1.4. *En criptografía, se utilizan los conceptos de* ***complejidad computacional*** *y* ***complejidad espacial*** *para medir los recursos requeridos*

por un algoritmo. La complejidad computacional mide el número de operaciones necesarias para ejecutar el algoritmo, mientras que la complejidad espacial mide la cantidad de memoria necesaria para su ejecución.
Ambos tipos de complejidad se pueden evaluar en función del peor de los casos o del caso promedio, dependiendo de la variabilidad del input. En este trabajo, nos enfocamos en la complejidad en el peor de los casos.
La complejidad computacional de un algoritmo, cuando se considera un conjunto de entradas I, se define como una función $f(n)$, donde $n = \log_2(|I|)$ es el tamaño del input en bits. Esta función mide el número máximo de operaciones que el algoritmo requiere para un input arbitrario. El número de operaciones se cuenta en términos de operaciones binarias, considerando sumas y multiplicaciones en $\mathbb{F}_2$.
Para expresar la complejidad, utilizamos la ***notación de Landau****, $O(g(n))$, que se define como sigue: dadas dos funciones f y g, ambas de $\mathbb{N}$ a $\mathbb{R}$, decimos que $f(n) = O(g(n))$ si existen constantes $N \in \mathbb{N}$ y $c \in \mathbb{R}_+$ tales que $f(n) \leq c \cdot g(n)$ para todo $n \geq N$.*

3.2. Problema del logaritmo discreto

El problema del logaritmo discreto, propuesto por Diffie y Hellman, marca el inicio de los criptosistemas de clave pública. Inicialmente, este problema se formuló en el contexto de cuerpos finitos $\mathbb{F}_p$. Sin embargo, con el tiempo se descubrió que su implementación en grupos de curvas elípticas sobre cuerpos finitos es más segura y computacionalmente más eficiente.
En primer lugar, definimos el problema del logaritmo discreto.

Definición 3.2.1. *Sea G un grupo cíclico arbitrario, generado por un elemento $g \in G$ y sea $a \in G \setminus \{0\}$. El* ***problema del logaritmo discreto*** *consiste en encontrar un exponente $n \in \mathbb{N}$ tal que $g^n = a$. En otras palabras, buscamos invertir la función $exp_g : \mathbb{Z}_N \longrightarrow G$, dada por $exp_g(n) = g^n$, donde $N = |G|$.*
Por tanto, buscamos calcular de manera explícita la función logaritmo discreto:

$$\log_g : G \longrightarrow \mathbb{Z}_N$$
$$g^n \longmapsto n$$

Nota. *La aplicación exp_g está bien definida, ya que si $|G| = N$, entonces $g^N = 1$, y por tanto, si $n = m + sN$, con $s \in \mathbb{Z}$, entonces $g^n = g^{m+sN} = g^m(g^N)^s = g^m, 1^s = g^m$, por lo que al tomar exponentes congruentes módulo N nos da el mismo resultado.*

La dificultad de hallar explícitamente el logaritmo discreto depende de cómo estén representados G y g.
Por las aplicaciones definidas en la definición 3.2.1, $G \cong \mathbb{Z}_N$. Si identificamos, bajo esa condición, $g \equiv 1 \mod N$, entonces el problema del logaritmo discreto sería trivial: si $a \in \mathbb{Z}_N$, $\log_g(a)$ es encontrar $n \in \mathbb{Z}_n$ tal que $1 \cdot n = a$ y por tanto $n = a$.

Por tanto, queremos usar grupos en los que no podamos obtener una representación sencilla de G y g, con el objetivo de evitar el problema trivial. En ese caso, no se conocen algoritmos de complejidad polinómica para resolver el problema del logaritmo discreto.
Existen dos grupos de gran importancia en este campo ya que encontrar en ellos el logaritmo discreto es muy complicado. Estos grupos son:

- El grupo multiplicativo de un cuerpo finito, $(\mathbb{F}_p^*, \cdot)$.
- Subgrupos cíclicos del grupo de puntos de una curva elíptica definida sobre un cuerpo finito.

Ahora bien, ¿por qué nos centraremos en usar subgrupos cíclicos del grupo de puntos de una curva elíptica definida sobre un cuerpo finito? Este tipo de grupos tiene una serie de propiedades que nos da ciertas ventajas a los basados en cuerpos finitos primos:

1. Proporcionan un alto nivel de seguridad con tamaños de clave más pequeños.
2. Mayor eficiencia computacional: en general la multiplicación de puntos en una curva elíptica es más rápida y requiere menos recursos computacionales en comparación con la multiplicación en grupos de cuerpos finitos.
3. Mayor resistencia a ataques (lo veremos en la sección 3.6).

Ejemplo 3.2.1. *Consideremos el grupo (multiplicativo)* $\mathbb{F}_{23}^*$. *Podemos escribir este grupo como el generado por el elemento* 3. *Tenemos que el logaritmo discreto de* 13 *en base* 3 *es* 5, *ya que* $13 \equiv 3^5 \mod 23$.

3.2.1. El problema del logaritmo discreto en curvas elípticas

Definición 3.2.2. *Sea* E *una curva elíptica definida sobre un cuerpo finito* $\mathbb{F}_p$, *y sean* $P, Q \in E(\mathbb{F}_p)$. *El* ***problema del logaritmo discreto sobre curvas elípticas*** *consiste en encontrar un entero* n *tal que* $Q = n \cdot P$, *donde la multiplicación está dada por la operación de grupos en la curva* E.
Este valor n *se denomina el logaritmo discreto de* Q *respecto de* P, *y se denota como* $n = \log_P(Q)$.

Antes de continuar, es importante hacer una pequeña anotación: en general, el logaritmo discreto sobre una curva elíptica de un punto puede no existir. Es decir, si tomamos dos puntos P y Q, puede suceder que Q no sea múltiplo de P. Por esta razón, en este trabajo solo consideramos puntos Q que pertenecen al subgrupo cíclico G generado por el punto P. De este modo, garantizamos que siempre existe el logaritmo discreto de Q respecto de P.
El algoritmo ingenuo de resolución del problema del logaritmo discreto en curvas elípticas es, dados $P, Q \in \mathbb{F}_p$, multiplicar secuencialmente $P, 2P, \cdots$ hasta hallar Q. Este método requiere hasta un máximo de N operaciones, siendo N el orden

de P, por lo que si utilizamos una curva elíptica donde P tenga un orden alto, el problema tiene una complejidad computacional alta.

Ejemplo 3.2.2. *Sea $E(\mathbb{F}_{23}) : Y^2 = X^3 + 9X + 17$. Tenemos los puntos $P = (3, 5), Q = (12, 6)$ en la curva. Vamos a ver cómo calcular el logaritmo discreto n de Q en base P.*
Una manera sería ir multiplicando P hasta que uno de sus múltiplos sea Q. Tenemos los siguientes múltiplos de P:
$P = (3, 5), 2P = (18, 10), 3P = (20, 20), 4P = (12, 6)$.
Por tanto, en este ejemplo, $n = 4$.

Nota. *Para este ejemplo, se utiliza el programa Sage, donde el producto de punto de la curva elíptica por escalar se encuentra ya definido.*

Es interesante mencionar una propiedad llamada compresión de puntos de una curva elíptica: esto consiste en reducir la cantidad de datos necesarios para representar los puntos en la curva, manteniendo la información esencial para realizar operaciones como la adición de puntos o la multiplicación por un escalar. En concreto, es muy utilizado en criptosistemas, ya que permite dado un punto $P = (x_0, y_0)$, enviar solamente la coordenada x y un bit adicional indicando qué valor de la raíz de y se debe escoger.

Como se abordará en la sección sobre ataques al problema del logaritmo discreto en curvas elípticas, resolver este problema en un grupo de curva elíptica sobre un cuerpo finito adecuado es computacionalmente inviable en un tiempo razonable. Esta dificultad es precisamente lo que hace que el método sea la base de numerosos criptosistemas actuales. A continuación, exploraremos algunos de estos criptosistemas.

3.3. Intercambio de claves de Diffie - Hellman

Primero, veamos en qué consiste y qué necesidad satisface este protocolo. Supongamos que un emisor y un receptor desean utilizar un cifrado privado y, para ello, necesitan compartir una clave secreta de manera segura. Sin embargo, no disponen de un canal de comunicación seguro que les permita hacerlo. Este problema de compartir una clave secreta de forma segura en un entorno inseguro parecía irresoluble hasta que, en 1975, Whitfield Diffie y Martin Hellman propusieron una solución revolucionaria utilizando el problema del logaritmo discreto.

El intercambio de claves propuesto por Diffie y Hellman permite a ambas partes generar una clave secreta compartida, sin necesidad de haberla comunicado previamente a través de un canal seguro.

El primer paso es escoger un grupo cíclico $G = \langle g \rangle$, sin ninguna restricción de privacidad sobre el mismo (será la clave pública). Veamos con esto el procedimiento que siguen ambos usuarios para conseguir la misma clave privada:

1. Tanto emisor como receptor deben de elegir un entero, $a \in \mathbb{Z}_N$ y $b \in \mathbb{Z}_N$, respectivamente, con $N = |G|$, que sólo sea conocido por ellos mismos.

2. Tanto emisor como receptor realizan el siguiente cálculo (respectivamente):

$$A = g^a, B = g^b.$$

3. Intercambian los resultados de ambas operaciones (a través de un canal no seguro a priori), de manera que ambos conocen A y B.

4. El último paso será para cada uno calcular:

$$A' = B^a, B' = A^b.$$

5. La clave final es el valor que comparten ($A' = B' = g^{ab}$).

Proposición 3.3.1. *En este punto, tanto emisor como receptor han obtenido el mismo valor, $A' = B'$.*

Demostración. $A' = B^a = (g^b)^a = g^{ab} = (g^a)^b = A^b = B'$. □

Ejemplo 3.3.1. *Usamos el grupo $\mathbb{F}_{23}$. Podemos usar $g = 5$, elemento primitivo del grupo. Supongamos que $a = 6$ y $b = 15$. Calculamos $A = 5^6 \equiv 8 \mod 23$, y por otro lado, $B = 5^{15} \equiv 19 \mod 23$. En este punto, los dos números calculados se intercambian y se vuelve a calcular: $A' = 19^6 \equiv 2 \mod 23$ y $B' = 8^{15} \equiv 2 \mod 23$.*

Definición 3.3.1. *Definimos el **problema de Diffie- Hellman** como, dado un grupo finito, G, y escogiendo $g \in G$, y dados g^a, g^b, encontrar g^{ab}.*

Corolario 3.3.1. *Si podemos resolver el problema del logaritmo discreto, podemos resolver el protocolo de intercambio de claves de Diffie - Hellman.*

Demostración. Por definición, para resolver el intercambio de claves tendríamos que hallar a, b, y para ello, bastaría calcular, sabiendo A, B, que son públicos, $\log_g A = a$ y $\log_g B = b$, que es precisamente resolver el problema del logaritmo discreto (en G). □

El principal problema de este protocolo es la imposibilidad de asegurar que A y B son los enviados por emisor y receptor: un criptoanalista podría interceptar uno de ellos, y mandar otro, B' por ejemplo, y de esta manera, tanto emisor como criptoanalista compartirían la clave: $g^{ab'}$. Este problema se solucionará más adelante usando firmas digitales, que veremos posteriormente.

3.3.1. Intercambio de claves de Diffie - Hellman en curvas elípticas

Como hemos visto, utilizar curvas elípticas sobre cuerpos finitos como grupo base para protocolos criptográficos nos da una serie de ventajas. Por ello veamos, cómo implementar el intercambio de claves de Diffie-Hellman para este grupo. El primer paso es escoger una curva elíptica $E(\mathbb{F}_p)$ y un punto en ella, P, que será público, de orden N grande (podríamos considerar el subgrupo generado por P, pero en este caso no es necesario). Veamos el procedimiento que siguen para obtener la misma clave privada:

1. Tanto emisor como receptor eligen un entero (que no se comparte), $n_A < N$ y $n_B < N$ respectivamente.
2. Tanto emisor como receptor realizan el siguiente cálculo:
$$Q_A = n_A P, Q_B = n_B P.$$
3. Los resultados de estas operaciones son intercambiados, de manera que ambos conocen: las claves públicas, su número secreto, Q_A y Q_B.
4. El último paso será, para cada uno, calcular:
$$Q'_A = n_A Q_B = (n_A n_B)P = Q'_B = n_B Q_A,$$
que es la clave privada que comparten finalmente.

Vemos un ejemplo:

Ejemplo 3.3.2. *Sea $p = 3851$ y E la curva de ecuación $E : Y^2 = X^3 + 324X + 1287$. Se escoge el punto de la curva $E(\mathbb{F}_{3851})$, $P = (920, 303)$. Se toman las siguientes claves privadas: $n_A = 1194$ y $n_B = 1759$, y calculamos: $Q_A = 1194P = (2067, 2178)$ y $Q_B = 1759P = (3684, 3125)$. Tras intercambiar los valores, calculamos la clave final: $n_A Q_b = 1194(3684, 3125) = (3347, 1242) = 1759(2067, 2178) = n_B Q_A$.*

De manera análoga al caso general, definimos el problema de Diffie- Hellman sobre un curva elíptica:

Definición 3.3.2. *Sea $E(\mathbb{F}_p)$ una curva elíptica sobre un cuerpo finito, y P, un punto de la curva. El **problema de Diffie- Hellman sobre un curva elíptica** es encontrar el valor de $n_1 n_2 P$, sabiendo únicamente $n_1 P$ y $n_2 P$.*

Corolario 3.3.2. *Como antes, resolver el problema del logaritmo discreto en curvas elípticas, podemos resolver el problema de Diffie - Hellman en curvas elípticas.*

Demostración. Esto es, ya que descifrar $Q'_A = Q'_B$ es equivalente a resolver $n_A P = Q_A$, sabiendo P y Q_A, que es precisamente el problema del logaritmo discreto sobre una curva elíptica. □

3.4. Criptosistema ElGamal

El Criptosistema de ElGamal fue el primer criptosistema de clave pública basado en el problema del logaritmo discreto.
Para poder ver el protocolo, primero debemos definir las claves. Escogemos un grupo cíclico $G = \langle g \rangle$, con $N = |G|$, y $n \in \mathbb{Z}_N \setminus \{0, 1\}$.

- Clave pública es la terna (G, g, g^n).
- Clave privada es el entero n.

El criptosistema consta de dos aplicaciones: una de cifrado y una de descifrado, que definimos a continuación. Dado $m \in G$,

- Cifrado
$$\begin{array}{rccc} e\colon & G & \longrightarrow & G^2 \\ & m & \longmapsto & (g^k, m.(g^n)^k) \end{array}$$
donde k se escoge de manera uniformemente aleatoria en $\mathbb{Z}_N$.

- Descifrado
$$\begin{array}{rccc} d_n\colon & e(G) \subseteq G^2 & \longrightarrow & G \\ & (g_1, g_2) & \longmapsto & g_2.g_1^{-n} \end{array}$$
El criptosistema funciona por la siguiente proposición:

Proposición 3.4.1. *Las aplicaciones de descifrado es inversa a la de cifrado.*

Demostración. Sea $m \in G$. Comprobamos explícitamente la condición. $d_n(e(m)) = d_n(g^k, mg^{nk}) = m.g^{nk}.g^{-nk} = m.$

□

Veamos un ejemplo:

Ejemplo 3.4.1. *Se escoge el grupo finito $\mathbb{F}_{23}^*$ y $g = 5$. Para realizar los cálculos se ha escogido un primo pequeño, pero en la realidad deberá ser grande. El emisor elige $a = 7$ como clave privada y calculamos la clave pública, $A \equiv 5^7 \equiv 17$* mód 23. *Sea $m = 15$ el mensaje. El receptor calculará con $k = 19$,*

$$c_1 \equiv 5^{19} \equiv 7 \mod 23, c_2 \equiv 15{,}17^{19} \equiv 6 \mod 23,$$

y se lo envía al emisor, que calcula:

$$x \equiv c_1^a \equiv 7 \mod 23, x^{-1} \equiv 14 \mod 23.$$

Por último, $x^{-1}c_2 \equiv 14{,}6 \mod 23 = 15 = m.$

Teorema 3.4.1. *Fijados un grupo y g y usados para encriptar un mensaje via ElGamal, y para el problema de Diffie Hellman, si un criptoanalista pudiera construir una aplicación para descifrar ElGamal eficientemente, podría usarlo para resolver el problema de Diffie- Hellman.*

Demostración. Sea $d_n : e(K) \to K$ una aplicación de descifrado para g^n, donde el criptoanalista conoce la clave pública. Ahora, para el problema de Diffie- Hellaman, el criptoanalista desea calcular g^{ab}, conociendo g^a y g^b. Entonces, lo que tiene que hacer el criptoanalista es escoger $g^n = g^a$, y pasar el mensaje cifrado (g^b, g_2), con g_2 aleatorio, por la aplicación d_n construida anteriormente, obteniendo $m = d_n(g_1, g_2) = g_2 g_1^{-n} = g_2 g^{-ab}$. Como se conoce m, g_2, puede obtener $g^{-ab} = mg_2^{-1}$, e invirtiendo, $g^{ab} = m^{-1}g_2$, que es lo que quería encontrar. □

3.4.1. Criptosistema de ElGamal en curvas elípticas

Veamos cómo podemos definir explícitamente el criptosistema de ElGamal sobre un grupo de curvas elípticas definidas sobre un cuerpo finito.
En primer lugar, se elige una curva elíptica sobre un cuerpo finito $E(\mathbb{F}_p)$ y un punto de la curva, P, de orden N grande (lo ideal sería que P generase el grupo). Definimos las claves:

- Clave privada: el receptor escoge $n_A \in \mathbb{Z}_N \setminus \{1\}$.
- Clave pública: el receptor calcula $Q_A = n_A.P$, y la clave pública que ambos conocen es la terna (E, P, Q_A).

Definimos la transmisión del mensaje, m del emisor al receptor.

- El emisor calcula el punto $M \in E(\mathbb{F}_p)$ asociado al mensaje m, y escoge un natural $k < N$ aleatorio, y codifica el mensaje de la siguiente manera:

$$C_1 = kP, C_2 = M + kQ_A.$$

- Envía el mensaje cifrado como el par (C_1, C_2).
- El receptor recupera el mensaje M, calculando:

$$C_2 - n_A C_1 = M.$$

Veamos que efectivamente, el criptosistema está bien definido.

Proposición 3.4.2. $C_2 - n_A C_1 = M$.

Demostración. $C_2 - n_A C_1 = (M + kQ_A) - n_A(kP) = M + k(n_A P) - n_A(kP) = M$. □

Este sistema criptográfico presenta algunas desventajas en comparación con ElGamal en grupos finitos de la forma $\mathbb{F}_p$:

1. El mensaje debe representarse como un punto de la curva elíptica, pero no existe un método directo y estándar para realizar esta transformación, lo que puede resultar complicado.
2. ElGamal sobre curvas elípticas requiere el doble de capacidad que ElGamal sobre grupos finitos de la forma $\mathbb{F}_p^*$. En ElGamal sobre grupos finitos de la forma $\mathbb{F}_p$, el mensaje es un valor m, con $2 < m < p - 1$, y el texto cifrado es un par de valores (c_1, c_2), con $2 < c_1, c_2 < p - 1$. Esto significa que, para cada mensaje, se necesitan dos valores cifrados. Sin embargo, cuando ElGamal se define sobre una curva elíptica, el mensaje $M \in E(\mathbb{F}_p)$ es un único punto de la curva, y según el teorema de Hasse, existen p posibles puntos. El texto cifrado, en este caso, es un par (C_1, C_2), que consta de cuatro valores, ya que cada punto en la curva tiene dos coordenadas.

Surge entonces la pregunta natural de si estas desventajas limitan el método o si existen soluciones para mitigarlas.

1. Para asociar un mensaje a un punto de la curva, una posible solución es elegir un punto M aleatorio y usarlo como si fuera el mensaje real. Esta técnica permite representar el mensaje como un punto de la curva, aunque con ciertas limitaciones.

2. Para solucionar el problema del tamaño del mensaje cifrado, se ha propuesto una técnica en la que solo se envía la primera coordenada de C_1 y C_2, acompañada de dos bits adicionales que proporcionan información extra (parecido a la técnica de compresión de puntos). El emisor debe calcular $C_1 - n_a C_1$, pero al conocer la primera coordenada de ambos puntos, puede calcular la segunda coordenada, sin necesidad de conocer el signo. Los dos bits adicionales permiten indicar el signo de la segunda coordenada de C_1 y C_2, lo que reduce la cantidad de datos enviados.

Ejemplo 3.4.2. *Sea E la curva elíptica de ecuación $y^2 = x^3 + 2011x + 1$ sobre $\mathbb{F}_{9765629}$, y sea $P = [0 : 1 : 1]$ un punto de E de orden 9765151.*
Supongamos que tenemos $m = 1733$ y por tanto $M = [1733, 1762, 1]$ es el mensaje original, $k_{priv} = 1756$ y por tanto, $k_{pub} = [959630 : 4100090 : 1]$. Usando el código que se puede encontrar en el Anexo, obtenemos que la encriptación de M es $[(197710 : 4520495 : 1), (6767712 : 839616 : 1)]$, y utilizando el algoritmo de decodificación, obtenemos de nuevo $[1733 : 1762 : 1]$.
Se puede encontrar el detalle del cálculo en el anexo.

Un caso particular es el algoritmo de Menezes-Vanstone:

Ejemplo 3.4.3. *Este criptosistema es derivado de ElGamal, pero introduce una serie de modificaciones que permiten corregir los problemas mencionados anteriormente.*
Igual que en el algoritmo general, se escoge una curva sobre un cuerpo finito $E(\mathbb{F}_p)$ y un punto de la misma, P.
El emisor elige una clave secreta, n_A, y calcula y envía $Q_A = n_A P$.
El receptor elige dos valores $m_1, m_2 \mod p$, y un número aleatorio, k, y calcula:

$$R = kP, S = kQ_A = (x_s, y_s).$$

Ponemos $c_1 \equiv x_s m_1 \mod p$ y $c_2 \equiv y_s m_2 \mod p$. Envía al emisor (R, c_1, c_2).
El emisor, con lo recibido calculará $T = n_A R = (x_T, y_T)$. Ponemos:

$$m_1' \equiv x_T^{-1} c_1 \mod p, m_2' \equiv y_T^{-1} c_2 \mod p.$$

Entonces, $m_1' = m_1$ y $m_2' = m_2$.

3.5. Criptosistema de Massey - Omura en curvas elípticas

Este criptosistema tiene la particularidad de que tanto el emisor como el receptor usan claves privadas. Encontrar una de las dos claves privadas consiste

en resolver el problema del logaritmo discreto.
Para este criptosistema sólo daremos el detalle del mismo definido sobre curvas elípticas.
Se considera una curva elíptica $E(\mathbb{F}_p)$ (siendo p el orden del subgrupo que consideramos) y $M \in E(\mathbb{F}_p)$ el mensaje a transmitir, ya asociado como punto de la curva. Definimos en primer lugar las claves:

- En este caso, tanto emisor como receptor escogen un par de claves de la forma (e_A, d_A) y (e_B, d_B) que cumplen la siguiente condición: $e_i d_i \equiv 1$ mód p.

A continuación, definimos la transmisión del mensaje:

1. El emisor envía $M_1 = e_A M$ al receptor.
2. El receptor envía $M_2 = e_B M_1$ al emisor.
3. El emisor calcula y envía al receptor $M_3 = d_A M_2 = d_A e_B e_A M = e_B M$.
4. Por último, el receptor calcula M a partir de $M = d_B M_3 = d_B e_B M$.

Para que el criptosistema esté definido correctamente, tenemos que probar que la condición definida en las claves, nos permite recuperar el mensaje.

Proposición 3.5.1. *$e_i d_i P = O$, para todo punto P de la curva $E(\mathbb{F}_p)$.*

Demostración. Haremos la prueba para e_A, d_A, siendo análogo para e_B, d_B.
Sabemos que $e_A d_A \equiv 1$ mód p, luego $e_A d_A = kp + 1$, para algún k. Sabemos que $pP = O$ para cualquier P en la curva, ya que p es el cardinal de la curva. Multiplicando por P aleatorio, obtenemos:

$$e_A d_A P = kpP + P = O + P = P.$$

Por tanto, $e_A d_A$ es el elemento neutro de la curva, O. □

La seguridad de este criptosistema se basa en que descifrar las claves es equivalente a resolver el problema de Diffie-Hellman.

3.6. Ataques al problema del logaritmo discreto

A lo largo de esta sección, se presentarán varios ataques al problema del logaritmo discreto en curvas elípticas: resolver este problema es en este grupo, en general, más complicado que sobre $\mathbb{F}_p^*$, de ahí la importancia y su uso en criptografía.
Veamos que existen ataques que funcionan de manera general, comentaremos por qué el Index Calculus no es válido para resolver el problema, lo que da importancia a los criptosistemas sobre curvas elípticas; y se presentará un ataque específico para criptosistemas implementados sobre un tipo concreto de curva elíptica.
Las principales referencias que se han usado en este capítulo han sido: [1] y [18].

3.6.1. Algoritmo ingenuo

Se basa en tratar de resolver el problema del logaritmo discreto en curvas elípticas calculando $nP, n = 0, 1, \ldots, p-1$ hasta llegar a $nP = Q$. De este modo, habríamos encontrado $n_i = \log_P(Q)$.
Este método no resulta factible si se escoge una curva definida sobre un cuerpo con p grande, ya que la complejidad es $O(p)$, exponencial.

3.6.2. Algoritmo Baby-Step Giant-Step

Consideramos una curva $E(\mathbb{F}_p)$ y un punto $P \in E(\mathbb{F}_p)$, de orden N. Consideraremos el subgrupo generado por P. Sea $Q \in \langle P \rangle$ tal que se quiere resolver $Q = nP$. El algoritmo Baby-Step Giant-Step consiste en lo siguiente:

1. En primer lugar, calculamos $m = \lfloor \sqrt{N} \rfloor$ y calculamos mP.
2. Para todo $i/0 \leq i \leq m$ calculamos iP, y se almacena en una lista.
3. Se calculan los puntos $Q - jmP, j = 0, 1, \ldots, m$ hasta que alguno coincida con algún valor de la lista del punto 2.
4. Si hemos hallado j tal que $iP = Q - jmP$, entonces $Q = iP + jmP = nP$, luego $n = i + jm$.

Como en el caso anterior, este método si N es grande es muy poco práctico, ya que requiere m pasos y su complejidad es $O(\sqrt{N})$. Además, necesitamos almacenar las dos listas de puntos calculadas en 2. y 3., por lo que se necesita además una complejidad espacial de $O(\sqrt{N})$.

Nota. *Para reducir la complejidad de este método, podemos guardar y comparar la mitad de los valores de i únicamente: Calculamos y almacenamos iP con $i \in [0, m/2]$ en el paso 2, y en el paso 3 comparamos $Q - jmP$ con $\pm iP$.*
De esta manera, tanto la complejidad espacial (necesitamos la mitad de almacenamiento de la lista), como la computacional (no hay que calcular iP con $i \in (m/2, m)$, sino $-iP$ con $i \in [0, m/2]$, que es más eficiente).

3.6.3. Algoritmo ρ de Pollard

Consideramos una curva $E(\mathbb{F}_p)$ y un punto $P \in E(\mathbb{F}_p)$, de orden N primo. Consideraremos el subgrupo generado por P. Sea $Q \in \langle P \rangle$ tal que se quiere resolver $Q = nP$.
El método consiste en en encontrar dos pares de enteros distintos, $(a, b), (a', b')$ módulo N tales que $aP + bQ = a'P + b'Q$. De esta manera, obtendríamos $n \equiv (a - a')(b' - b)^{-1} \mod N$.
Surge la pregunta de cómo podemos seleccionar los pares; existen dos métodos:

- Una primera opción es seleccionar aleatoriamente $a, b \in [0, N-1]$. A continuación, se calcula $aP + bQ$, y se guarda la terna $(a, b, aP + bQ)$.
  Se repite este proceso, escogiendo pares aleatorios y guardando las ternas

hasta que obtengamos otro par distinto, (a', b') tal que $aP + bQ = a'P + b'Q$.
En esta caso, se habrá resuelto el problema, ya que tendremos $(a-a')P = (b'-b)Q$, y usando la definición inicial de $Q = nP$. Juntando ambas partes, $(a - a')P = (b' - b)nP$ y como consecuencia, $n = (a - a')(b' - b)^{-1}$.

Nota. *La desventaja de este método es que además de tener complejidad computacional $O(\sqrt{N})$, también la tiene espacial, $O(\sqrt{N})$.*

- La segunda opción, parte de la idea de reducir la complejidad espacial. Para ello, se define una función:

$$f_{a,b} : \langle P \rangle \longrightarrow \langle P \rangle,$$

tal que dados $a, b \in [0, \dots N-1]$ y $R \in \langle P \rangle$ de la forma $R = aP + bQ$, podamos calcular $fa.b(R) = R'$, con R' de la forma $R' = a'P + b'Q$, con $a', b' \in [0, \dots, N-1]$.
Para ello, dividimos $\langle P \rangle$ en L particiones de igual tamaño, $S_1, \dots, S_L$.
A continuación, definimos otra función, $H : \langle P \rangle \longrightarrow [1, L]$, tal que $H(X) = j$ si $X \in S_J$. Con esto elegimos $a_j, b_j \in [0, N-1]$ de manera aleatoria para $j \in [1, L]$. La función que buscábamos estará definida por:

$$f(R) = R + a_j P + b_j Q,$$

con $j = H(X)$. Así, si para R se cumple $R = aP + bQ$ también, $f(R) = R' = a'P + b'Q$, con $a' = a + a_j \mod N$ y $b' = b + b_j \mod N$.
Como el conjunto de puntos generados por P es finito, encontraremos la igualdad para $(a, b) \neq (a', b')$, que es lo que necesitábamos.

3.6.4. Algoritmo de Pohlig- Hellman

Este algoritmo es factible cuando el orden de P, N, satisface $N = n_1^{e_1} . \cdots . n_t^{e_t}$, con n_i primos y $n_i^{e_i}$ pequeños, ya que se basa en reducir el problema calculando los logaritmos discretos en los subgrupos de orden primo de $\langle P \rangle$ (utilizando otros métodos que pueden ser eficientes en subgrupos de tamaño pequeño). Como antes, queremos resolver $Q = nP$, con P, Q en la curva $E(\mathbb{F}_p)$.
Este algoritmo se basa en el Teorema chino de los restos, que recordamos a continuación:

Teorema 3.6.1. *Sean $n_1, \dots, n_k \in \mathbb{N}$ primos entre sí y sean $a, \dots, a_k \in \mathbb{Z}$. Entonces, el sistema de congruencias:*

$$x \equiv a_i \mod n_i, \forall i \in \{1, \dots, k\}$$

tiene una única solución módulo $N = \prod_{i=1}^{k} n_i$.

Podemos ver la prueba (donde además hallamos la solución) en [4].
En nuestro caso, tenemos $N = n_1^{e_1} \cdots n_t^{e_t}$ y las congruencias de la forma:

$$n_i \equiv n \mod n_i^{e_i},$$

para cada $i \in [1, t]$. El Teorema Chino de los Restos nos asegura que este sistema tiene una única solución, que es precisamente el valor del logaritmo discreto que se quiere calcular.
A continuación se muestra cómo calcular a_i; detallamos el proceso para el cálculo de a_1.
Escribamos $a_1 \equiv \alpha_0 + \alpha_1 n_1 + \alpha_2 n_1^2 + \cdots + \alpha_{e_1-1} n_1^{e_1-1} \mod n_1^{e_1}$ con $\alpha_i \in [0, n_1 - 1]$, la representación de a_1 en base n_1. A continuación, calculamos una lista de valores, $T_i = \{j(\frac{N}{n_i}P/0 \leq j \leq n_i - 1\}$. Por otro lado, calculamos $\frac{N}{n_1}Q = \alpha_0(\frac{N}{n_1}P + (\alpha_1 + \alpha_2 n_1 + \dots)nP = \alpha_0 \frac{N}{n_1}P$.
Comparando estos valores con los de la lista T_1, el común será α_0.
Para calcular α_1, consideramos $Q_1 = Q - \alpha_0 P$, y volvemos a utilizar los pasos anteriores, tendremos $\frac{N}{e_1^2}Q_1 = (a_1 + a_2 n_1 + \dots)\frac{N}{n_1}P = a_1 \frac{N}{n_1}P$, multiplicando la forma en base n_1 de a_1 por $\frac{N}{n_1^2}$. El valor de a_1 es aquel que coincide con alguno de los valores de T_1.
Recursivamente, calculamos los coeficientes α_k para hallar a_1.
Repitiendo el mismo proceso, calculamos los a_i con $i = \{1, \dots, t\}$. Aplicando el teorema chino de los restos, hallamos el resultado.
La complejidad de este algoritmo es $O(\sqrt{p})$, donde p es el mayor primo que divide a N. Por tanto, este método es útil si los n_i que dividen al orden son pequeños.

3.6.5. Index Calculus

Este método es uno de los más eficaces a la hora de resolver el problema del logaritmo discreto sobre el grupo multiplicativo de $\mathbb{F}_p$, ya que intenta reducir el problema a resolver un sistema de ecuaciones lineales. De forma resumida, para resolver $g^x \equiv n \mod p$, se escoge una serie de primos (que se considerará como una base) $S = \{p_1, \dots, p_n\}$ y calculamos el logaritmo discreto de cada g^{p_i} de manera que podamos expresar el problema inicial como una combinación lineal de los logaritmos discretos calculados, y lo podamos resolver en forma de un sistema de ecuaciones.
EL punto crítico es poder tener una lista S de primos, lo más corta posible, y que el cálculo del logaritmo discreto sobre ellos sea rápido, sin perder la capacidad de que las combinaciones de los elementos de S como producto sean lo más amplias posibles, para poder hallar el logaritmo buscado.
Sin embargo, aunque este método sea eficaz para grupos $\mathbb{F}_p^*$, su complejidad en curvas elípticas es exponencial, por lo que no resuelve el problema en un tiempo óptimo: esto es por la dificultad para factorizar elementos del grupo de curvas elípticas y la falta de definición de un equivalente a los primos en este grupo.
Debido a esta resistencia, los criptosistemas realizados sobre el grupo de curvas elípticas son más seguros que los realizados sobre $\mathbb{F}_p^*$. Sin embargo, no son completamente inmunes a este ataque, y en la actualidad se están desarrollando métodos para aplicarse en curvas elípticas definidas sobre extensiones de cuerpos, aunque todavía con baja eficiencia, mediante el algoritmo de Semaev: en curvas definidas sobre $\mathbb{F}_{2^n}$, para resolver el problema de las descomposición de

factores primos, se trata de encontrar $P_1, \ldots, P_{n-1}$ puntos en $E(\mathbb{F}_{2^k})$, tales que $P_n = \sum_{i=1}^{n-1} P_m$ y utilizar bases de Gröbner. Esto sobrepasa los objetivos del trabajo, por lo que no entramos en ello, podemos encontrar más detalle en [13].

3.6.6. Ataque por emparejamiento: Ataque MOV

Todos los ataques vistos anteriormente se basan en utilizar ciertas propiedades del grupo para tratar de calcular el logaritmo discreto. Sin embargo, existen ataques válidos para algunos tipos de curvas elípticas.
Exiten varios ataques que aprovechan las propiedades de un tipo de curva específico; en este trabajo veremos el ataque MOV (Menezes-Okamoto-Vanstone) para las curvas supersingulares. Utilizamos como referencia [18].
Este ataque utiliza el emparejamiento de Weil para trasladar el problema del logaritmo discreto en una curva elíptica a uno sobre, $\mathbb{F}_{p^k}$, donde es más probable que el problema sea resuelto (recordamos que k es el grado de extensión de la curva).
Sea E una curva sobre $\mathbb{F}_p$, donde necesitamos resolver el problema del logaritmo discreto $Q = nP$, con $P, Q \in E(\mathbb{F}_p)$ tales que, el orden de P es un primo (distinto de p), $l > \sqrt{p}+1$ y Q un múltiplo de P. Lo primero a tener en cuenta es calcular k, como el grado de extensión de E respecto de l.
Veamos en primer lugar la idea detrás de este algoritmo, que es precisamente el motivo por el que funciona:

Teorema 3.6.2. *Sea E una curva sobre $\mathbb{F}_p$, y $P, Q \in E(\mathbb{F}_p)$ con l el orden del punto P. Entonces, existe $n \in \mathbb{Z}$ tal que $Q = nP$ si y sólo si $lQ = O$ y $e_l(P, Q) = 1$.*

Demostración. $\Longleftarrow$) Supongamos que se da $lQ = O$ y $e_l(P, Q) = 1$. Por definición de punto de torsión, tenemos que $Q \in E[l]$ y como l y p son primos, $MCD(p, l) = 1$. Por el teorema 2.3.3, $E[l] \cong \mathbb{Z}_l \times \mathbb{Z}_l$. Elegimos un punto $R \in E[l]$ tal que $\{P, R\}$ forman una base de $E[l]$. Con esto, escribimos $Q = aP + bR$, con $a, b \in \mathbb{Z}$. Calculando el emparejamiento de Weil de P y R, $e_l(P, R) = \chi$, donde χ es una raíz l-ésima de la unidad.
Como por hipótesis tenemos que $e_l(P, Q) = 1$, por tanto desarrollando y con las propiedades del emparejamiento, obtenemos que $e_l(P, Q) = e_l(P, aP + bR) = e_l(P, P)^a . e_l(P, R)^b = 1 . e_l(P, R)^b = e_l(P, R)^b = \chi^b$, luego $1 = \chi^b$.
Por ser χ una raíz l-ésima de la unidad, $b \equiv 0 \mod l$ y por tanto $bR = O$, luego como $Q = aP + bQ = aP$.

$\Longrightarrow$) Supongamos $Q = nP$. Entonces, multiplicando por el orden de P, l, obtenemos $lQ = nlP = nO = O$.
Además, $e_l(P, Q) = e_l(P, P)^n = 1^n = 1$.

□

Describimos a continuación el algoritmo MOV:

1. En primer lugar, calculamos el número de puntos de la curva $E(\mathbb{F}_{p^k})$, con k el grado de extensión, $N = |E(\mathbb{F}_{p^k})|$.

Notemos que dado que P es de grado l y pertenece a la curva $E(\mathbb{F}_p)$, entonces $l|N$.

2. Escogemos aleatoriamente un punto $T \in E(\mathbb{F}_{p^k}) \setminus E(\mathbb{F}_p)$, y calculamos $T' = \frac{N}{l}T$. Si este valor es O,volvemos a seleccionar otro T.

3. Calculamos los emparejamientos de Weil :

$$x = e_l(P, T') \in \mathbb{F}^*_{p^k}$$

y

$$y_2 = e_l(Q, T') \in \mathbb{F}^*_{p^k}.$$

4. Aquí llegamos al paso clave, que es calcular el problema del logaritmo discreto:

$$y = x^n$$

en el grupo multiplicativo $\mathbb{F}^*_{p^k}$.
El entero n que resuelve el problema en el grupo multiplicativo, 4 resuelve también el problema original 3.6.6.

Nota. *Las curvas supersingulares son muy sensibles a este ataque ya que no tienen puntos de l-torsión, por lo que su grupo de torsión es reducido y más fácil de calcular.*

3.7. Firmas digitales

En esta sección, vamos a utilizar el grupo de curvas elípticas para resolver otro tipo de problema: autenticar la identidad del emisor de un "documento"; es decir, firmas digitales. Veamos una breve introducción a este problema.
Los cuatro elementos básicos y necesarios para resolver este problema usando firmas digitales son: una clave privada (para firmar), K_{priv}, una clave pública (para verificar), K_{pub}, un algoritmo de firma (con el documento a firmar, D, y K_{priv} se genera un nuevo documento firmado, D^{sig}) y un algoritmo de verificación (con D, D^{sig}y K_{pub} devuelve verdadero si D^{sig} es una firma de D asociada a $_{priv}$, y falso en caso contrario).
A continuación se incluye un esquema general del proceso de autentificación:

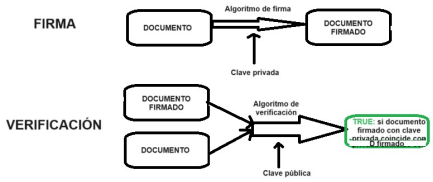

Figura 3.2: Esquema firmas digitales

Es importante aclarar que las firmas digitales no son aleatorias y deben cumplir con ciertos requisitos básicos:

1. La firma debe estar vinculada al mensaje: La firma es única y está asociada específicamente al documento. Si D^{sig} es el documento D firmado, no puede ser utilizada para otro documento diferente, D'.

2. La firma es intransferible: La firma de una persona no puede ser falsificada ni replicada por otra persona.

3. La firma es verificable: Cualquier persona, de manera pública, puede verificar la autenticidad de la firma.

4. La firma no puede ser rechazada por el firmante (S): Una vez que el documento ha sido firmado, el firmante no puede negar ni rechazar su firma en el futuro.

Estos requisitos nos proporcionan la información necesaria para seleccionar el tipo de criptosistema adecuado para generar los algoritmos de firmas digitales. Dado que la firma debe ser personal y privada, mientras que su verificación debe ser pública, los criptosistemas más adecuados para este propósito son los basados en clave pública.
Dado D un conjunto de documentos, en general no se firmará todo el conjunto, ya que su tamaño puede ser grande, sino solo una parte. En estos casos se usará la llamada función de Hash:

Definición 3.7.1. *La **función de Hash** es una función sobreyectiva, $h : D \to C$, donde el cardinal de C es mucho menor que el de D. En general, usaremos $C = \{0,1\}^k$.*

Nota. *La función de Hash es pública. Por este motivo, no puede obtenerse D de $h(D)$.*
Esto es porque sino un criptoanalista podría replicar la firma.

Veamos el proceso general de la firma digital. Supongamos que S genera un par de claves, $(K_{\text{priv}}, K_{\text{pub}})$, para un criptosistema de clave pública específico, junto con los algoritmos de cifrado, $c_{K_{\text{pub}}} : M \to C$, y de descifrado, $d_{K_{\text{priv}}} : C \to M$, que satisfacen las siguientes propiedades:

$$d_{K_{\text{priv}}}(c_{K_{\text{pub}}}(m)) = m \quad \forall m \in M,$$

$$c_{K_{\text{pub}}}(d_{K_{\text{priv}}}(c)) = c \quad \forall c \in C.$$

1. FIRMA:
   En primer lugar, se aplica la función de hash al documento, obteniendo $h(D)$. Se utiliza a continuación la clave privada de S para firmar el hash, obteniendo $D_s = d_{K_{priv}}(h(D)) \in M$. Entonces, S envía D_s, a la persona que verifica la firma (aunque en ciertos casos se envía el par, (D, D_s)).

2. VERIFICACIÓN:
   Se comprueba la veracidad de la firma calculando $c_{K_{pub}}(D_s)$ y verificando si el resultado es igual a $h(D)$:

   $$c_{K_{pub}}(D_s) = c_{K_{pub}}(d_{k_{priv}}(h(D)).$$

   Si la firma es válida, esto debe coincidir con $h(D)$.

Comprobamos que este proceso es de una firma digital:

Proposición 3.7.1. *El proceso definido anteriormente, con la función de Hash, cumple los requisitos para ser firma digital.*

Demostración.
1. La firma es única: $h(D) \neq h(D')$ si $D \neq D'$, y como la función $d_{k_{priv}}$ tiene inversa, por lo que es biyectiva y se cumple que $D_s = d_{k_{priv}}(h(D)) \neq d_{k_{priv}}(h(D')) = D'_s$.

2. La firma es intransferible: la clave privada es sólo conocida por S, y es muy difícil calcularla a partir de la clave pública (al ser un criptosistema de clave pública).

3. Públicamente verificable: Como hemos dicho antes, $h(D)$ es público, así como D_s y la clave pública, que son los elementos necesarios para comprobar $c_{k_{pub}}(D_s) = c_{k_{pub}}(d_{k_{priv}}(h(D))$.

4. La firma no puede ser rechazada: $h(D), D_s, k_{pub}$ se usan para verificar D_s una vez publicada, por lo que no puede ser rechazada.

□

Introduciremos las firmas digitales con RSA para poder a continuación aplicarlas a curvas elípticas. Se utiliza como principal referencia [10].

3.7.1. Firmas digitales con RSA

Para el proceso de firma digital con RSA, se escogen dos números primos grandes, p y q, distintos entre sí, y se define $N = pq$, con el valor de N público. A continuación, se escogen dos enteros $s, v \in \{1, \ldots, (p-1)(q-1)\}$, tales que $sv \equiv 1$ mód $(p-1)(q-1)$. Llamaremos a s el exponente de firma, que corresponde a la clave privada, y a v el exponente de verificación, que corresponde a la clave pública.
Veamos el proceso de firma y verificación para esta firma, para un documento D, tal que $1 < D < N$.

1. FIRMA: El firmante calcula $S \equiv D^s$ mód N y lo envía al receptor.
   En caso de utilizar una función de hash pública, primero calcula $h(D)$ y luego envía la firma $d_s = h(D)^s$.

2. VERIFICACIÓN: La persona que verifica la firma calcula S^v mód N, y si coincide con D mód N entonces la firma es válida.
   En caso de usar una función de Hash, calculamos D_s^v, y si coincide con $h(D)$ mód N, es válida.

Proposición 3.7.2. *Si la firma es correcta para un par de claves* $(K_{pub} = v, K_{priv} = s)$*, entonces* $S^v \equiv D$ mód N.

Demostración. Por el teorema de Euler , Hoffstein ([9]), $S^v \equiv D^{sv} \equiv D$ mód N. En caso de usar la función de Hash, $h(D)^{sv} \equiv h(D)^{vs} \equiv h(D)$, por lo que $D_s^v \equiv h(D)^s v \equiv h(D)$ mód N. □

Veamos un pequeño ejemplo, con números pequeños por simplificar los cálculos.

Ejemplo 3.7.1. *Sea* $N = 15$*, con* $p = 3$ *y* $q = 5$*, y se escoge* $s = 3, v = 3, s, v \in \{1, \cdots, 8\}$*. Sea* $D = 10$.
La firma en este caso sería $10^3 \equiv 10$ mód 15*, y se lo envía al verificador.*
Éste, calcula $10^3 \equiv 10$ mód 15*. Como* $D = 10$*, la firma es correcta pues coincide módulo 15.*

3.7.2. Firma DSA (Digital Signature Algorithm)

El siguiente paso antes de ver cómo realizar firmas digitales en curvas elípticas es analizar el DSA.
Para esta firma, se escogen dos primos, p, q tales que $q|p-1$. En la práctica, normalmente se tiene que $1000 < log_2(p) < 2000$ y $160 < log_2(q) < 320$, así que q será mucho más pequeño que p.
Una vez seleccionados los dos primos, el firmante escoge $g \in \mathbb{F}_p^*$ de orden q; por ejemplo, siendo α un elemento primitivo de $\mathbb{F}_p$, podemos escoger $g = \alpha^{(p-1)/q}$.
El siguiente paso para el firmante es escoger la clave privada, $s \in \mathbb{N}$, y publica (p, q, g, v), con $v \equiv g^s$ mód p. Esta cuaterna es la clave pública.
Veeamos el proceso de firma y verificación de la firma de este método, para un documento D, tal que $1 \leq D \leq q$.

1. FIRMA: Se considera $e/1 \leq e \leq q$ aleatorio y se calcula:

$$S_1 = (g^e \mod p) \mod q; S_2 = (D + sS_1)e^{-1} \mod q.$$

   La firma digital que se envía es el par (S_1, S_2).

2. VERIFICACIÓN: La persona que verifica la firma calcula:

$$V_1 \equiv DS_2^{-1} \mod q; V_2 \equiv S_1 S_2^{-1} \mod q.$$

   La firma es válida si $(g^{V_1} v^{V_2} \mod p) \mod q = S_1$.

Comprobamos la validez de la firma:

Proposición 3.7.3. *Si la firma DSA es válida, entonces* $(g^{V_1} v^{V_2} \mod p) \mod q = S_1$.

Demostración. Tenemos $(g^{V_1} v^{V_2} \mod p)$. Como $V_1 \equiv DS_2^{-1}$ y $V_2 \equiv S_1 S_2^{-1}$ y $v \equiv g^s$, entonces $(g^{V_1} v^{V_2} \mod p) \equiv g^{DS_2^{-1}} g^{sS_1 S_2^{-1}} \equiv g^{(D+sS_1)S_2^{-1}} \mod p \equiv g^e \mod p$, teniendo en cuenta que $S \equiv (D + sS_1)e^{-1}$.
Con esto, tomando módulo q, $(g^{V_1} v^{V_2} \mod p) \mod q \equiv (g^e \mod p) \mod q = S_1$, como queríamos demostrar. □

3.7.3. ECDSA (Elliptic Curve Digital Signature Algorithm)

Una vez visto el proceso de firma DSA sobre el grupo $\mathbb{F}_p$, nos interesa cambiar el grupo de definición de la firma a una curva elíptica, ya que aumentamos la seguridad de la firma, como veremos en esta sección. De esta manera, basándose en el DSA, surge el ECDSA que comentaremos a continuación.
Consideramos $E(\mathbb{F}_q)$ una curva elíptica, con q primo o potencia de 2, y P un punto de la curva de orden N. Para generar las claves, el firmante escoge un entero, $s \in 1, \cdots, N-1$, y a continuación calcula y comparte $V = sP$. La clave privada (de firma) es s y la pública (de verificación), es V.
Veamos el proceso de firma y verificación de este método, para un documento D, sin mayor reestricción ya que usaremos una función de Hash.

1. FIRMA: en primer lugar, utilizando una función de Hash pública, siendo D el documento a firmar, se calcula $h(D)$.
   El firmante escoge un entero aleatorio, e tal que $1 \leq e \leq N-1$ y calcula $eV = (x, y)$. A continuación, calcula $k \equiv x \mod N$, cambiando tantas veces de e hasta que $k \neq 0$. Una vez se da esta condición, calcula el inverso de e, $e^{-1} \mod N$
   Por último, calculamos $S \equiv e^{-1}(h(D)+k) \mod N$. Si $S = 0$, necesitamos escoger otro entero e, hasta que $S \neq 0$.
   El firmante envía la terna (D, k, S).

2. VERIFICACIÓN: la persona que verifica recibe (D, k, S), tiene acceso a la función de Hash y conoce el punto P y el orden de la curva, además de la clave pública.

Hay que tener cuidado con que los enteros recibidos k, S estén en el intervalo $[1, N-1]$, ya que si no cumplen esta condición, el verificador no puede asegurar nada.
Para comprobar la firma, en primer lugar se calcula $h(D)$, y calculamos $w \equiv S^{-1} \mod N$, y a continuación, $u_1 = h(D)w \mod N$ y $u_2 = kw \mod N$.
Por último, calculamos es $Q = u_1P + u_2V$. Dependiendo del valor de Q tenemos dos casos:

a) Si $Q = O$, la firma se rechaza.

b) En caso contrario, si $Q = (x, y)$, calculamos $v = x \mod N$. Si $v = k$, la firma es válida. Sino, se rechaza.

Probemos la validez de la firma:

Proposición 3.7.4. *Si la firma es válida, entonces $v = k$.*

Demostración. Sea D el documento y (k, S) el par que recibe la persona que verifica.
Entonces, $S \equiv e^{-1}(h(D) + sk) \mod N$.
Despejando, $e \equiv S^{-1}(h(D)+sk) \equiv S^{-1}h(D)+S^{-1}sk \equiv wh(D)+wsk \equiv u_1+u_2s \mod N$, ya que $w \equiv S^{-1}$ y $u_2 \equiv kw \mod N$.
Por tanto, $Q = u_1P + u_2V = (u_1 + u_2s)P = eP$, luego $v = k$. □

¿Por qué se elige el método de firma ECDSA antes que el RSA?
La principal ventaja del ECDSA es que el tamaño de claves del mismo es mucho menor que el del RSA, para el mismo nivel de seguridad. En la siguiente tabla, que encontramos en Levy, [11], podemos ver el número de bits necesario para las dos claves, para el mismo nivel de seguridad:

Longitud de las claves RSA (bits)	Longitud de las claves ECDSA (bits)
1024	192
2048	256

Cuadro 3.1: Longitud de las claves del RSA contra la de ECDSA

La ventaja de la reducción del tamaño de claves es que el tiempo de generación de firmas y ejecución de la firma es mucho menor, lo que ayuda a reducir almacenamiento.
Otra ventaja de usar el ECDSA es que para que las firmas no sean reproducibles (es decir, si un criptoanalista consigue firmas de varios mensajes, no pueda replicarla para uno nuevo), es necesario que el problema del logaritmo discreto sea difícil de resolver en el grupo. Como hemos visto, éste es más difícil de resolver en el caso de curvas elípticas.
Como decíamos en la definición 3.7.1, si la funicón de Hash no es segura, un criptoanalisata podría falsificar la firma: escogiendo $t \in \mathbb{Z}$ aleatoriamente y calculando $Q + tP = (x, y)$, podría poner $S = x$ y calcular $e = xt \mod N$. Si

consiguiera un mensaje D tal que $e = H(D)$, entonces el par (x, S) es una firma válida de D.

Ejemplo 3.7.2. *Podemos ver un ejemplo de firma digital sobre la curva elíptica definida por* $y^2 = x^3 + 980 * x + 1$ *sobre* $\mathbb{F}_{1031}$ *en el anexo de los algoritmos.*

Bibliografía

[1] Dr Tun Myat Aung y Ni Ni Hla. «A study of general attacks on elliptic curve discrete logarithm problem over prime field and binary field». En: (2017).

[2] John William Scott Cassels. *LMSST: 24 Lectures on Elliptic Curves*. 1991.

[3] C Castillo. «Curvas Elípticas». En: (2017).

[4] Thomas H Cormen et al. *Introduction to algorithms*. 2022.

[5] William Fulton. «Algebraic curves». En: (2008).

[6] Andreas Gathmann. «Plane algebraic curves». En: (2018).

[7] Menezes Hankerson. *Guide to Elliptic Curve Cryptography*. 2004.

[8] Harzo Hida. «Arithmetic of Curves». En: (2011).

[9] Jeffrey Hoffstein. «An Introduction to Mathematical Cryptography». En: 2008.

[10] Don Johnson, Alfred Menezes y Scott Vanstone. «The elliptic curve digital signature algorithm». En: (1999).

[11] Sharon Levy. «Performance and Security of ECDSA». En: (2015).

[12] Alfred J Menezes. *Elliptic curve public key cryptosystems*. 1993.

[13] Igor Semaev. «Summation polynomials and the discrete logarithm problem on elliptic curves». En: (2004).

[14] Joseph H Silverman. *The arithmetic of elliptic curves*. 2009.

[15] Joseph H Silverman, Jill Pipher y Jeffrey Hoffstein. *An introduction to mathematical cryptography*. 2008.

[16] Joseph H Silverman y John Torrence Tate. *Rational points on elliptic curves*. 1992.

[17] Janet Visser. «Schoof's algorithm: Point counting on elliptic curves». Tesis doct. 2020.

[18] Lawrence C Washington. *Elliptic curves: number theory and cryptography*. 2008.

[19] MAEVE COATES WELSH. «ELLIPTIC CURVE CRYPTOGRAPHY». En: (2017).

Capítulo 4

Code

En esta sección mostramos los cálculos y el proceso que seguimos en los ejemplos vistos:

4.1. ElGamal

Para el criptosistema ElGamal de nuestro ejemplo:

```
#Definimos la curva eliptica E: y^2 = x^3 + 2011*x + 1 sobre el cuerpo
#finito F_9765629
p = next_prime (9765628);p

E_a_b = EllipticCurve(GF(p),[2011,1] )
P = E_a_b ([0,1]) #Punto [0,1,1]

n = P.order()
#Clave publica, privada
k_priv = 1756
k_pub = k_priv * P
#Mensaje como punto de la curva
m= 1733
M = E_a_b.lift_x( m )
#Encriptacion del mensaje

k = randint( 1 , n - 1 )
C_1 = k * P
C_2 = M + k * k_pub
print(f"Mensaje cifrado: ({C_1}, {C_2})")

#Descifrado del mensaje
MD= C_2 - k_priv * C_1
print(f"Mensaje descifrado: ({MD})")
```

print("El mensaje descifrado es igual al original?", MD == M)

Y tenemos el siguiente resultado:

```
Mensaje cifrado: ((5722153 : 3456149 : 1), (2762308 : 8967199 : 1))
Mensaje descifrado: ((1733 : 1762 : 1))
¿El mensaje descifrado es igual al original? True
```

Figura 4.1: ElGamal

4.2. Firmas digitales

Para (firmas digitales):

```
#Creamos la curva eliptica y^2 = x^3 +980x +1 en el cuerpo F_1031
p = next_prime (1030)
F_p = GF( p )
E_a_b = EllipticCurve( GF ( p ) , [980 ,1] )
#P es un punto aleatorio de la curva. Calculamos su orden
P = E_a_b.random_element()
n = P.order()
#s es la clave privada y Q es la clave publica
s= 34567890987654321;
Q = s * P;
#Generacion de la firma
e = 17 #Mensaje que queremos firmar (realmente seria el hash)
ok = False
while not ok : #Generamos una firma valida segun el algoritmo visto
#en la seccion de firmas
      k = randint( 1 , n - 1 )
      x = ( k * P )[ 0 ]
      u = Integer( x )
      r = u % n
      if r != 0 :
          S = ( inverse_mod( k , n ) * ( e + s * r ) ) % n
          if S != 0 :
              ok = True
#s es la firma (valida) que se ha generado
#Verificacion de la firma
w = inverse_mod( S , n )
R = e * w * P + r * w * Q
x = R[ 0 ]
u = Integer( x )
v = u % n
print("La firma recibida es valida?", v == r)
```

```
¿La firma recibida es válida? True
```

Figura 4.2: Firmas digitales

4.3. Puntos de la curva

Además, he usado el siguiente código para hallar puntos en la curva que necesitase:

```
    # Definir p como el siguiente primo despues de 9765628
p = next_prime(9765628)
print(f"El valor de p es: {p}")

# Paso 2: Crear la curva eliptica E sobre el campo GF(p)
E = EllipticCurve(GF(p), [2011, 1])

# Buscamos puntos en base a valores de x
x
for x in range(1, 10000):
    rhs = (x^3 + 2011*x + 1) % p
    # Lo tendriamos que adaptar si cambiamos los valores de la curva


    if rhs.is_square():
    # Si es un cuadrado, podemos calcular y
        y = rhs.sqrt()
        P = E(x, y)  # Creamos el punto P
        print(f"El punto P es: {P}")

 # Calculamos el orden del punto P
        orden_P = P.order()
        print(f"El orden de P es: {orden_P}")
    else:
        print("El valor no es un cuadrado en F_p")
```