Curvas elípticas y aplicaciones a la criptografía

Pablo Vicioso Escorza

Trabajo de fin de grado en Matemáticas
Universidad de Zaragoza

Director del trabajo: Miguel Ángel Marco Buzunáriz
28 de junio de 2021

Abstract

In this project, two difference but related fields will be studied. On the one hand we study elliptic curve theory and on the other hand we show some elliptic curve applications to cryptography. Every main topic is going to be explained briefly and the most important results will be remarked.

Elliptic curve theory

First of all, we explain algebraic preliminaries about curves in the projective plane. These concepts are shown in order to understand all concepts we use later, then we can introduce *elliptic curves* defined over a general field. These elliptic curves are cubics in the projective plane without singular points.

After this, we prove that an elliptic curve can be expressed with the projective general Weierstrass equation. This projective equation can be dehomogenized. Once we have this, when the characteristic of the field si different of 2 and 3,then this dehomogenized Weierstrass general equation can be transformed into the affine equation: $y^2 = x^3 + ax + b$, which is the standard equation to work with elliptic curves.

Then, an operation over elliptic curves points known as *the chord-and-tangent method* is explained. This is a geometric method to produce, given two points on an elliptic curve,a third one. This operation is naturally conmutattive and we prove the existence of a neutral element and an inverse element.

The next goal will be to prove the associativity. In order to show this, we need to introduce the concept of *pencil of curves*. A main theorem which characterizes the existence of a pencil having 8 differrent points of the projective plane is proven. The proof of the associativity is based on the construction of a pencil with 9 different points that are the intersection points of two different cubics.

Lemma. *The set of points of an elliptic curve forms an abelian group with the operation defined by the chord-and-tangent method.*

After this, elliptic curves defined over finite fields are studied. Now the main objective will be prove the *Hasse's Theorem* which bounds the number of rational points that an elliptic curve defined over a finite field has.

Theorem (Hasse). *The order of an elliptic curve $E(\mathbb{F}_q)$, $|E(\mathbb{F}_q)|$, verifies*

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

The proof of this theorem involves various concepts. We start defining an *endomorphism* of elliptic curves. Some key concepts such as the *degree* or *separability* of endomorphisms are defined. After this, some principal results about endomorphisms are proven: the relationship between the degree and the kernel of an endomorphism and some properties about *Frobenius endomorphism*.

One surprising fact in the proof of Hasse's theorem is the use of a topological property of rational numbers, a characteristic zero field, in order to prove a finite fields result.

An easy example is given in order to show the importance of this theorem.

Elliptic curve cryptography

We start mentioning two hard problems which some public key schemes are based on are named. These two problems are the integer factorization problem and the discrete logarithm problem. Elliptic curve cryptography is based in this second one.

Next, we find a comparison of key lenght needed to obtain the same level of security between a protocol using a finite field and the same protocol using the group generated by an elliptic curve defined over a finite field.

Given G, P, two points of an elliptic curve. The number k will be said to be the *discrete logarithm* of P respect to G if $P = kG$. The *elliptic curve discrete logarithm problem(ECDLP)* is based on the difficulty to find k knowing G and P. Having this explained, we will see a way to generate keys and how to compute kG efficiently.

Classical elliptic curve cryptography. At the beggining of this part, the classical cryptographic protocols based on elliptic curvas will be explained along with a brief note remarking cryptographic hash functions. These protocols are named Diffie-Hellman and ElGamal. The first one allows two people to agree on a point a point in the elliptic curve safely and the second one allows to send an encrypted message. The security of this protocols is based on the ECDLP.

There exist other schemes which allow a party to sign a digital document and a second party to verify the validity of the signature. These schemes are named *Digital signatures*. In this case, we will explain digital signatures with elliptic curves.

The most famous digital signature scheme with elliptic curves is *ECDSA*. This scheme will be explained along with remarks about its security. Another digital signature scheme is explained, *Schnorr Signatures*. Schnorr signatures are linear. This property allows many people to sign a digital document with only one signature. This is called *Key Aggregation for Schnorr Signatures* or *Multiple Signatures*. The linearity of Schnorr Signatures is a great advantage over ECDSA, which are not linear.

Elliptic curve cryptography nowadays An important function of elliptic curves today is its use in crypto coins, such as Bitcoin, who uses ECDSA to sign its transacctions.

But, Bitcoin has a huge problem of storage space because it stores the whole information of its transactions. *Mimblewimble* is a new protocol whose purpose is to solve this problem. Mimblewimble does not store all the information because of its transactions allows a person to prove the validity of a coin only knowing the current state of the coin and its origin.

Mimblewimble does not use scripts (like Bitcoin), therefore all the transactions have to be done with signatures. One could think that this will bring many limitations, but some of these limitations can be overcome thanks to a new scheme of signatures, called *Adaptor Signatures*.

Adaptor Signatures are a modification of Multiple Signatures. This modification will allow two parties to sign digital documents without trusting each other, because if one party signs the transaction, then some secret information must be revealed to the other party.

Índice general

Abstract III

1. Introducción y objetivos 1

2. Preliminares algebraicos 3

3. Curvas elípticas 5
3.1. Ecuación de Weierstrass 5
3.2. Estructura de grupo de las curvas elípticas 6
3.2.1. Asociatividad del grupo asociado a una curva elíptica 7
3.3. Curvas elípticas sobre cuerpos finitos 10

4. Criptografía sobre curvas elípticas 15
4.1. Protocolos criptográficos 16
4.2. Esquemas de firma digital con curvas elípticas 17
4.2.1. Firmas Schnorr 19
4.3. Implementación de las curvas elípticas en la actualidad 21

Bibliografía 25

Capítulo 1

Introducción y objetivos

Contexto

Desde hace varios siglos, las curvas elípticas han sido estudiadas por los matemáticos y se han utilizado para resolver un amplio abanico de problemas como, por ejemplo, el *Último Teorema de Fermat*.

Sin embargo, la aplicación de las curvas elípticas en la criptografía vino mucho después. En el año 1985, Neal Koblitz y Victor Miller propusieron independientemente utilizar las curvas elípticas para diseñar sistemas criptográficos de clave pública. Esto empezó a cobrar una mayor relevancia a finales de la década de los 90, cuando los sistemas con curvas elípticas comenzaron a recibir una gran aceptación comercial y compañías privadas incluyeron dichos protocolos en sus productos de seguridad.

Uno de los principales problemas al que se enfrenta la criptografía con curvas elípticas, y los principales criptosistemas de clave pública, son los ordenadores cuánticos. En 1994, Peter Shor presentó un algoritmo para ordenadores cuánticos capaz de calcular logaritmos discretos y factorizar enteros, principales problemas matemáticos en los que se basa la criptografía de clave pública, de forma eficiente. Sin embargo, hasta la fecha no se ha podido construir un ordenador cuántico con la capacidad suficiente para resolver instancias de estos problemas no triviales.

Problemas abordados

Los problemas abordados en el presente trabajo han sido por un lado el estudio teórico sobre las curvas elípticas así como el estudio de las mismas definidas sobre cuerpos finitos y, por otro lado, el estudio y la explicación de diversos protocolos criptográficos que hacen uso de las curvas elípticas.

En el estudio teórico, daremos primero ciertos preliminares algebraicos necesarios para la comprensión de las curvas elípticas. Se pasará a continuación a introducir las mismas y a la explicación de una operación que podemos realizar sobre ellas, que las dotará de una estructura de grupo. Por último, ya que se suelen utilizar curvas elípticas definidas sobre cuerpos finitos en criptografía, proporcionaremos una estimación sobre el número de puntos de una curva elíptica definida sobre un cuerpo finito (*Teorema de Hasse*).

En el estudio de los protocolos criptográficos basados en curvas elípticas, empezaremos abordando e introduciendo el problema del logaritmo discreto y las ventajas que poseen las curvas elípticas sobre otros grupos a la hora de llevar a cabo dichos protocolos. Después se explicarán algunos de los protocolos clásicos más utilizados y acabaremos dando algunas de las últimas aplicaciones que se han desarrollado en los últimos años y que se continúan implementando.

Comentario. Nótese que un estudio exhaustivo sobre la teoría de curvas elípticas es inviable en un trabajo de esta extensión, debido a la enorme amplitud de dicha teoría. Es por esto por lo que se han elegido algunos de los contenidos más representativos de esta teoría en relación a su aplicación en la criptografía.

Objetivos

Los objetivos que se persiguen al realizar este trabajo son:

- Explicar la aritmética de los puntos de las curvas elípticas.
- Realizar un estudio en profundidad de las curvas elípticas sobre cuerpos finitos, con el fin de dar una aproximación del número de puntos que posee una curva elíptica sobre un cuerpo finito.
- Explicar el problema del logaritmo discreto sobre curvas elípticas y exponer protocolos criptográficos clásicos cuya seguridad resida en la dificultad de resolver dicho problema.
- Desarrollar los principales protocolos de firmas digitales con curvas elípticas.
- Mostrar un protocolo de criptomonedas actual que utilice curvas elípticas en sus transacciones.

Capítulo 2

Preliminares algebraicos

2.0.1 Definición. Sea $\mathbb{F}$ un cuerpo. El plano afín sobre $\mathbb{F}$, que denotaremos $\mathbb{A}^2$ es el conjunto de puntos de $\mathbb{F}^2 = \mathbb{F} \times \mathbb{F}$. Se define el *plano proyectivo* sobre $\mathbb{F}^2$, $\mathbb{P}^2 := (\mathbb{F}^3 \setminus \{0\})/\sim$, donde $\sim$ es la relación de equivalencia dada por

$$(x,y,z) \sim (x',y',z') \Leftrightarrow \exists \lambda \in (\mathbb{F} \setminus \{0\}) \text{ tal que} (x,y,z) = \lambda(x',y',z')$$

.

Cada una de las clases de equivalencia se llama punto proyectivo y lo denotaremos por$(x : y : z)$. Dado un punto proyectivo $(x : y : z)$ sabemos que $x \neq 0$ o $y \neq 0$ o $z \neq 0$. Los puntos proyectivos con $z = 0$ forman lo que se denomina *recta del infinito*. De manera análoga definimos el *espacio proyectivo* de dimensión $n, \mathbb{P}^n$.

2.0.2 Definición. Dado un polinomio $F(x_1,...,x_n) \in \mathbb{F}[x_1,...,x_n]$ diremos que es un *polinomio homogéneo* si todos sus monomios tienen el mismo grado.

Notar que si $x_{n+1} \neq 0$, $(x_1 : ... : x_n : x_{n+1}) \in \mathbb{P}^n \Rightarrow (x_1 : ... : x_{n+1}) \sim (\frac{x_1}{x_{n+1}} : ... : \frac{x_n}{x_{n+1}} : 1)$, entonces podemos tomar $x_{n+1} = 1$ y cambiar el resto de coordenadas. Podemos tomar cualquier otra coordenada para este razonamiento y para los que van a ser explicados a continuación. Ahora podemos ver la relación entre el espacio afín y el espacio proyectivo. Consideramos estas aplicaciones:

$$\begin{gathered} \mathbb{A}^n \to \mathbb{P}^n \\ (a_1,...,a_n) \mapsto (a_1 : ... : a_n : 1) \\ \left(\frac{a_1}{a_{n+1}},...,\frac{a_n}{a_{n+1}}\right) \leftarrow (a_1 : ... : a_n : a_{n+1}) \end{gathered}$$

Donde cabe notar que la segunda aplicación está definida solamente en la parte del espacio proyectivo cuyos puntos tienen la última coordenada no nula.

Relación de polinomios: A estos procesos los llamaremos *homogeneización*$(()^*)$ y *deshomogeneización*$(()_*)$:

$$\begin{aligned} ()^* : \mathbb{F}[x_1,...,x_n] &\to \mathbb{F}[x_1,...,x_{n+1}] \\ f &\mapsto f^* = x_{n+1}^d f\left(\frac{x_1}{x_{n+1}},...,\frac{x_n}{x_{n+1}}\right), \text{ donde } d \text{ es el grado de } f. \\ ()_* : \mathbb{F}[x_1,...,x_{n+1}] &\to \mathbb{F}[x_1,...,x_n] \\ F &\mapsto F_* = F(x_1,...,x_x,1) \end{aligned}$$

Tenemos entonces $(f^*)_* = f$ y $x_{n+1}^d (F_*)^* = F$. A partir de ahora tomaremos x_{n+1} como la variable que homogeneiza y deshomogeneiza los polinomios.

A los polinomios homogéneos los llamaremos *formas*.

A continuación vamos a definir las dos aplicaciones siguientes:

$$\begin{aligned} I : \mathscr{P}(\mathbb{A}^n) &\to \mathscr{P}(\mathbb{F}[x_1,...,x_n]) \\ T &\mapsto I(T) = \{f \in \mathbb{F}[x_1,...,x_n] | f(p) = 0 \ \ \forall p \in T\} \\ V : \mathscr{P}(\mathbb{F}[x_1,...,x_n]) &\to \mathscr{P}(\mathbb{A}^n) \\ S &\mapsto V(S) = \{p \in \mathbb{A}^n | f(p) = 0 \ \ \forall f \in S\} \end{aligned}$$

2.0.3 Definición. Diremos que un subconjunto de $\mathbb{A}^n$ es un *conjunto algebraico afín* si es de la forma $V(S)$ para algún $S \subset \mathscr{P}(\mathbb{F}[x_1,...,x_n])$. Notar también que $V(S) = V(I(S))$ donde $I(S)$ es el ideal generado por S.

De manera análoga al caso afín, podemos definir variedades algebraicas proyectivas e ideales homogéneos, tomando puntos proyectivos del espacio proyectivo $\mathbb{P}^n$ y homogeneizando los polinomios de los ideales tal y como se ha marcado anteriormente.

2.0.4 Definición. Sean $f(x,y) \in \mathbb{F}[x,y]$ y $F(x,y,z) \in \mathbb{F}[x,y,z]$ no constantes. Entonces el conjunto $\mathscr{C}(f) = \{(x,y) | f(x,y) = 0\}$ se denomina *curva afín* y $\mathscr{C}(F) = \{(x:y:z) | F(x:y:z) = 0\}$ *curva proyectiva*.

Llamaremos *grado* de una curva proyectiva $\mathscr{C} = \mathscr{C}(F)$ al grado del polinomio homogéneo F. Llamaremos *rectas* a las curvas proyectivas de grado 1, *cónicas* a las curvas proyectivas de grado 2 y *cúbicas* a las curvas proyectivas de grado 3.

2.0.5 Definición. Sea $\mathscr{C} = \mathscr{C}(F) = \{(x:y:z) | F(x:y:z) = 0\}$ una curva proyectiva y $p = (a:b:c) \in \mathscr{C}$. Decimos que p es un *punto singular* de $\mathscr{C}$ si satisface las ecuaciones:

$$\begin{cases} \frac{\partial f}{\partial x}(p) = 0. \\ \frac{\partial f}{\partial y}(p) = 0. \\ \frac{\partial f}{\partial z}(p) = 0. \end{cases}$$

Con esto, decimos que una curva es *lisa* si todos sus puntos son no singulares. De este modo podemos definir la *multiplicidad* de p en $\mathscr{C}$ como el menor entero m_p tal que

$$\frac{\partial^m F}{\partial^i x \partial^j y \partial^k z}(p) \neq 0 \qquad \text{para algunos i,j,k cumpliendo i+j+k=m}$$

Sea $\mathscr{D} = \mathscr{D}(G)$ otra curva proyectiva, llamamos curva intersección a $\mathscr{C} \cap \mathscr{D}$.

2.0.6 Definición. Sea $\mathscr{C} = \mathscr{C}(F)$ una curva proyectiva, llamamos *recta tangente a $\mathscr{C}$ en p* a la recta dada por la ecuación $F_x(p)x + F_y(p)y + F_z(p)z = 0$.

2.0.7 Proposición. Sean $\mathscr{C}$, $\mathscr{D}$ curvas proyectivas sin componentes comunes. Entonces

$$m_p(\mathscr{C} \cap \mathscr{D}) \geq m_p(\mathscr{C}) \cdot m_p(\mathscr{D}).$$

Además, se obtiene la igualdad $\Leftrightarrow$ las tangentes a $\mathscr{C}$ en p son distintas de las tangentes en $\mathscr{D}$ en p.

2.0.8 Lema. Sea $F \in F[x,y,z]$. Entonces se puede descomponer F en producto de factores irreducibles $F = F_1^{e_1} \cdots F_s^{e_s}$. Con esta descomposición, podemos escribir la curva como unión de sus componentes irreducibles: $\mathscr{C}(F) = \mathscr{C}(F_1) \cup \cdots \cup \mathscr{C}(F_s)$.

2.0.9 Definición. Una curva $\mathscr{C}$ se dice irreducible si $\mathscr{C} = \mathscr{C}(F)$ con F irreducible.

2.0.10 Teorema (Teorema de Bezout)**.** Sean F y G formas de grado n y m respectivamente, y $\mathscr{C} = \mathscr{C}(F), \mathscr{D} = \mathscr{D}(G)$ curvas proyectivas sin componentes comunes. Entonces

$$\sum_{p \in \mathscr{C} \cap \mathscr{D}} m_p(\mathscr{C} \cap \mathscr{D}) = nm = gradF \cdot gradG.$$

Capítulo 3

Curvas elípticas

3.0.1 Definición. Una curva elíptica es una curva lisa $E \subset \mathbb{P}^2$ de grado 3.

3.1. Ecuación de Weierstrass

Cualquier curva elíptica puede ser escrita en $\mathbb{P}^2$ como una cúbica de la siguiente forma:

$$Ax^3 + Bx^2y + Cx^2z + Dxyz + Ey^2z + Fy^2x + Gy^3 + Hz^3 + Iz^2x + Jz^2y = 0.$$

Tomando un sistema de referencia adecuado y haciendo el cambio de variable correspondiente, tales curvas se pueden expresar en la forma proyectiva de Weierstrass:

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

O en el plano afín, deshomogeneizando la ecuación anterior nos queda, que una *curva elíptica E* se define por una ecuación de Weierstrass de la forma:

$$E : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6, \tag{3.1}$$

donde $a_1, a_2, a_3, a_4, a_6 \in \mathbb{F}$. El conjunto de puntos de E son $E(\mathbb{F}) = \{\infty\} \cup \{(x,y)\ \mathbb{F} \times \mathbb{F} | y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6\}$, donde ∞ es el punto del infinito de la curva.

Diremos que E *está definida sobre* $\mathbb{F}$ y lo denotaremos $E(\mathbb{F})$. A $\mathbb{F}$ lo llamaremos *cuerpo base*. En ocasiones denotaremos a la curva elíptica simplemente como E cuando se sobreentienda el cuerpo base en el que está definida.

3.1.1 Proposición. Sea $E(\mathbb{F})$ una curva elíptica. Si $char(\mathbb{F}) \neq 2, 3$, entonces el cambio de variables

$$(x,y) \mapsto \left(\frac{x - 3a_1^2 - 12a_2}{36}, \frac{y - 3a_1x}{216} - \frac{a_1^3 + a_1a_2 - 12a_3}{24} \right)$$

transforma E en la curva

$$y^2 = x^3 + ax + b, \tag{3.2}$$

donde $a, b \in \mathbb{F}$.

Demostración. Es suficiente comprobar el cambio de variable. □

Comentarios:

1. Dado el uso que vamos a hacer de las curvas elípticas, de aquí en adelante vamos a desarrollar el trabajo con la ecuación de Weierstrass simplificada (3.2), asumiendo pues que $char(\mathbb{F}) \neq 2, 3$.
2. Si $char(\mathbb{F}) = 2$, la ecuación anterior (3.2) no es válida ya que tiene puntos singulares.
3. Si $char(\mathbb{F}) = 3$, la ecuación (3.2) es válida, pero existen curvas que no tienen esta forma.

3.2. Estructura de grupo de las curvas elípticas

En este apartado vamos a dotar de estructura de grupo a una curva elíptica. Con este fin, vamos a definir una operación en $E(\mathbb{F})$, mediante el *método de la cuerda y la tangente* del siguiente modo:

- Tomamos E una curva elíptica dada por la ecuación de Weierstrass y sea $\mathscr{O}$ un punto de ella, al que llamaremos *punto base*.
- Sean $P, Q \in E$ y L la recta que pasa por estos dos puntos.
- Sea ahora R el tercer punto de corte de L con E; es decir, $E \cap L = \{P, Q, R\}$. Si $P = Q$, entonces L' (recta tangente a E en P) corta a E en un segundo punto; es decir, $E \cap L' = \{P, P, R\}$.
- Ahora volvemos a repetir el mismo proceso tomando los puntos $R, \mathscr{O}$. Llamamos $\overline{L}$ a la recta que pasa por $R, \mathscr{O}$ y que cortará a E en otro tercer punto S, así $E \cap \overline{L} = \{R, \mathscr{O}, S\}$. Si $\overline{L}$ es tangente a E en $\mathscr{O}$, entonces $\overline{L}$ no corta a E en un tercer punto y tomamos $S = \mathscr{O}$.
- Entonces denotamos $S := P \oplus Q$.

A continuación se muestra un ejemplo gráfico del método de la curva y la tangente:

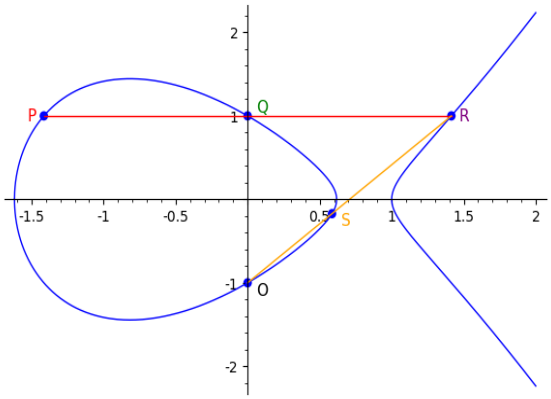

Figura 3.1: Ejemplo método de la cuerda y la tangente. Curva elíptica: $y^2 = x^3 - 2x + 1$ definida sobre $\mathbb{R}$

3.2.1 Proposición (Propiedades del método de la cuerda y la tangente)**.**

1. Conmutatividad:

$$P \oplus Q = Q \oplus P.$$

2. Existencia de elemento neutro:

$$P \oplus \mathscr{O} = P.$$

3. Existencia de elemento opuesto:

$$\text{Dado } P \in E \Rightarrow \exists (-P) \text{ tal que } P \oplus (-P) = \mathscr{O}.$$

Demostración. 1. Inmediata, ya que solo hay una recta que corte a P y Q.

2. Podemos ver la existencia de elemento neutro de forma directa ya que: la recta L que pasa por P y $\mathscr{O}$ corta a la curva en un tercer punto R. Ahora para hallar $P \oplus \mathscr{O}$ tomamos la recta que pasa por R y $\mathscr{O}$, que corta a la curva de nuevo en P y así $P \oplus \mathscr{O} = P$.

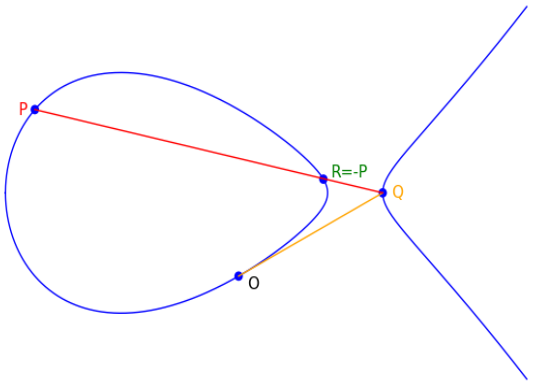

Figura 3.2: Aclaración demostración existencia elemento opuesto. Curva elíptica: $y^2 = x^3 - 2x + 1$ definida sobre $\mathbb{R}$

3. Tomamos $P \in E$ y L la recta tangente a $\mathscr{O}$. Está recta cortará a E en otro punto que llamamos Q; es decir, $E \cap L = \{\mathscr{O}, \mathscr{O}, Q\}$. Tomamos ahora la recta que une P y Q, L', que cortará a E en otro tercer punto que llamamos R, pero nos percatamos que $P \oplus R = \mathscr{O}$ por lo que ya hemos encontrado el punto que denotamos por $-P = R$.

   Una aclaración geométrica de esta demostración se puede ver en la Figura 3.2.

□

3.2.2 Comentario.

- Si tomamos como elemento neutro $\mathscr{O}$ un punto de inflexión de la curva, entonces se cumple que:
  - Si tenemos A, B, C tres puntos alineados de la curva, entonces $A \oplus B \oplus C = \mathscr{O}$. Esto se debe a que el punto $A \oplus B$ estará alineado con C y $\mathscr{O}$ y por ser este último de inflexión la recta tangente a él no volverá a cortar a E.
- Hemos tomado para la demostración de las anteriores propiedades el punto base de la curva $\mathscr{O}$ como un punto cualquiera de la misma pero, si vemos la curva en forma de Weierstrass, entonces tomaremos generalmente como punto base el punto del infinito de la curva, $\mathscr{O} = (0 : 1 : 0)$, que además es un punto de inflexión de nuestra curva. De este modo, dados dos puntos de la curva, P, Q y L la recta que pasa por estos puntos, entonces esta recta corta en otro tercer punto R, y si ahora queremos calcular la intersección de la recta que pasa por R y $\mathscr{O}$ con la curva elíptica, basta con encontrar el simétrico de R respecto al eje de las abscisas (reflexión de R).

  En criptografía, se suele escoger este punto ya que entonces la suma de dos puntos es mucho más rápida, lo cual nos proporciona una mayor eficiencia computacional.

Una vez vistas estas propiedades, solo falta ver la asociatividad de esta operación para poder afirmar que $(E, \oplus)$ es un grupo abeliano, donde $\oplus$ es la operación que acabamos de definir.

3.2.1. Asociatividad del grupo asociado a una curva elíptica

Ahora vamos a probar que si dotamos a una curva elíptica de la operación anterior mediante el método de la cuerda y la tangente, entonces para todo punto $P, Q, R \in E$ se tiene que $(P \oplus Q) \oplus R = P \oplus (Q \oplus R)$. Veamos a continuación las herramientas matemáticas para probarlo:

3.2.3 Definición. Llamaremos $\mathbb{P}_d$ a las curvas en $\mathbb{P}^2$ de grado d.

3.2.4 Lema. Las curvas de grado d en el espacio $\mathbb{P}^2 = \mathbb{F}[x, y, z]$ constituyen un espacio proyectivo de dimensión $\frac{d(d+3)}{2}$

Demostración. Llamamos V_d al espacio vectorial con base el conjunto de monomios de grado d en x,y,z. El número de monomios es $\#\{(i,j,k)|i,j,k \geq 0, i+j+k=d\}$. Se puede probar que este número es $\binom{d+2}{d}$; es decir

$$dim\{F|F\text{forma de grado d en}\mathbb{F}[x,y,z]\} = \binom{d+2}{d} = \frac{(d+2)(d+1)}{2} = \frac{d^2+3d+2}{2},$$

de donde

$$dim\ \mathbb{P}_d = \frac{d^2+3d+2}{2} - 1 = \frac{d(d+3)}{2}.$$

□

De esto, concluimos directamente que podemos interpretar las formas de grado 3 como el espacio $\mathbb{P}_3$, que tendrá dimensión 9. De esta forma cada punto en $\mathbb{P}^9$ representa una única curva de grado 3 en $\mathbb{P}^2$ y cada curva de grado 3 en $\mathbb{P}^2$ se corresponde a un único punto en $\mathbb{P}^9$.

3.2.5 Definición. Denotamos por *sistema lineal de curvas* a un subespacio proyectivo de $\mathbb{P}_d$. Llamamos *haz de curvas* a un sistema lineal de dimensión 1.

3.2.6 Lema. Sea $\Lambda \subset \mathbb{P}_d$ un espacio lineal de dimensión mayor o igual que 1 entonces son equivalentes:

1. Λ es un haz.
2. $\exists\ a \in \mathbb{P}^2$ tal que hay una única curva en Λ que pase por a.
3. $\exists\ a, a' \in \mathbb{P}^2$ tal que no existe ninguna curva en Λ que pase por ambos.

Demostración. 1) $\Rightarrow$ 2) Tomamos una curva en Λ, y podemos tomar $a \in \mathbb{P}^2$ por el que no pase la curva. Sea $\mathscr{C}_a$ el conjunto de curvas de grado d que pasan por a. Notar que $\mathscr{C}_a$ es un subespacio proyectivo de codimensión 1 ya que estas curvas tan solo cumplen una condición. Entonces es claro que $\Lambda \not\subset \mathscr{C}_a$, luego $\Lambda \cap \mathscr{C}_a$ es un único punto de $\mathbb{P}_d$, por tanto existe una única curva de Λ que pasan por a.

2) $\Rightarrow$ 3) Tomamos ahora un punto por el que no pasa la única curva en Λ que pasa por a.

3) $\Rightarrow$ 1) Sea $\mathscr{C}_{a'}$ el conjunto de las curvas de grado d que pasan por a'. $\mathscr{C}_{a'}$, al igual que $\mathscr{C}_a$ es un subespacio proyectivo de codimensión 1, por lo tanto $\mathscr{C}_a \cap \mathscr{C}_{a'}$ tiene a lo sumo codimensión 2 . Si se diera que Λ tuviera dimensión mayor o igual que 2, su intersección con cualquier subespacio proyectivo de codimensión 2 sería no vacía, lo cual significaría que existe una curva en Λ que pasa por a y a', pero esto sería una contradicción con la hipótesis. □

3.2.7 Teorema. Dados ocho puntos $a_i \in \mathbb{P}^2$ distintos y Λ el conjunto de cúbicas pasando por todos ellos. Entonces, Λ es un haz si y sólo si no existen cónicas que pasan por los 8 puntos y no existen rectas que pasan por 5 de los 8.

Demostración. Lo haremos en diferentes casos con $dim\ (\Lambda) \geq 1$, pues es la intersección de 8 hiperplanos en $\mathbb{P}_3$ de dimensión 9. Por lo que acabamos de ver, sabemos que Λ es un haz si y sólo si existe un punto $a \in \mathbb{P}^2$ tal que existe una única cúbica en Λ que pasa por a o, equivalentemente, existen dos puntos a, a' tales que no hay cúbicas en Λ que pasen por ambos.

- Si existe una cónica C que contenga a los 8 puntos, entonces $C \cup L$ con cualquier $a \in L$ es una cúbica en Λ que pasa por a.

  A partir de ahora asumiremos que no existen cónicas que contengan a los 8 puntos.

- Si existe una recta L que contenga a cinco puntos (salvo reordenación $a_1, \cdots, a_5$) cada vez que tomemos una cónica $C \ni a_6, a_7, a_8$ tenemos una cúbica $L \cup C$ en Λ. Como hay infinitas cónicas que pasan por a_6, a_7, a_8, a tenemos infinitas cúbicas en Λ que pasan por a, luego Λ no es un haz.

- Si existe una recta $L \ni a_1, \cdots, a_4$ y $a_5, \cdots, a_8 \notin L$ entonces, por el Teorema de Bezout (2.0.10) una cúbica en Λ será $L \cup C$, con C cónica cumpliendo $a_5, \cdots, a_8 \in C$. Por otro lado, notar que no existe $L' \ni a_5, \cdots, a_8$, pues entonces $L \cup L'$ sería una cónica cumpliendo $a_1, \cdots, a_8 \in L \cup L'$, que no se da en este caso porque hemos demostrado antes que no había cónicas conteniendo los 8 puntos. Ahora si tomamos $a \notin L$ ni en ninguna recta que contenga a 3 de los puntos $a_5, \cdots, a_8$. Entonces existe una única cónica C que pasa por $a_5, \cdots, a_8, a$ entonces $L \cup C$ es la única cúbica en Λ que pasa por a. Así, Λ es haz en este caso.

- Si existe $L \ni a_1, a_2, a_3$ y $a_4, \cdots, a_8 \notin L$. Existe una única cónica $C \ni a_4, \cdots, a_8$. El punto a lo buscamos en $L \setminus \{a_1, a_2, a_3\}$, la cúbica corta a la recta en 4 puntos, luego las cúbicas en Λ que pasan por a serán $L \cup C'$ donde $a_4, \cdots, a_8 \in C'$, $C' = C$. Sólo hay una tal cúbica, luego Λ también es un haz en este caso.

  A partir de ahora también podemos suponer que no hay tres puntos $a_1, \cdots, a_8$ alineados.

- Si existe una cónica $C \ni a_1, \cdots, a_7$ entonces es irreducible (ya que no pueden ser dos rectas) y corta a una cúbica de Λ en 7 puntos, pero entonces debe tener con ella una componente en común, pero como C es irreducible debe ser $C \cup L$ con $a_8 \in L$. Con tomar $a \neq a_8, a \notin C$ ya tenemos que la recta L es única, y por tanto la cúbica. Así, de nuevo Λ es un haz en este caso.

- Supongamos que existe una cónica $C \ni a_1, \cdots, a_6$ y $a_7, a_8 \notin C$. Ahora vamos a exigir $a \in C, a \neq a_1, \cdots, a_6$ y cualquier cúbica en Λ que pasa por a ha de tener con C una componente común. Por la misma observación de antes C es irreducible. De nuevo la cúbica es $C \cup L$, y debe ser $L = \langle a_7, a_8 \rangle$. Tenemos unicidad y Λ es un haz.

- Este es el caso más general. Empezamos tomando la única cónica $C \ni a_1, \cdots, a_5$. Como los puntos no están alineados la cónica es irreducible. Tomamos $a, a' \in C, a, a' \neq a_1, \cdots, a_5$. Cualquier cúbica en Λ que pase por a y a' debe compartir alguna componente irreducible con C, que es irreducible, luego será $C \cup L$ donde L es una recta. Nos falta imponer que a_6, a_7, a_8 estén en la cónica. Pero $a_6, a_7, a_8 \notin C$, luego $a_6, a_7, a_8 \in L$. Pero esto contradice la hipótesis de que no hay tres puntos alineados, luego por la segunda caracterización de los haces, Λ es un haz.

□

3.2.8 Corolario. Sea $\mathscr{C} = \mathscr{C}(F) \subset \mathbb{P}^2$ cúbica irreducible y $\{a_1, \ldots, a_8\}$ 8 puntos distintos en $\mathscr{C}$. Entonces, el sistema lineal de cúbicas que pasan por todos ellos es un haz.

Demostración. Como $\mathscr{C}$ es una cúbica irreducible, por el teorema de Bezout, si L es una recta que corta a la cúbica, $\sum_{p \in \mathscr{C} \cap L} m_p(\mathscr{C} \cap L) = 3 \cdot 1 = 3$. Por otro lado, si C es una cónica que corta a $\mathscr{C}$, de nuevo por el teorema de Bezout $\sum_{p \in \mathscr{C} \cap C} m_p(\mathscr{C} \cap C) = 2 \cdot 3 = 6$, entonces por el Teorema anterior 3.2.7. □

3.2.9 Corolario. Sean $\mathscr{C} = \mathscr{C}(F)$, $\mathscr{D} = \mathscr{D}(G)$ dos cúbicas que se cortan en 9 puntos. Entonces cualquier cúbica que pase por 8 de ellos, pasa también por el noveno.

Demostración. Por el Corolario anterior, las cúbicas que pasan por 8 de los nueve puntos forman un haz. Los elementos del haz tienen ecuación $t_0 F + t_1 G$. Como F y G se anulan en los 9 puntos, todas las cúbicas del haz pasan por los 9 puntos. □

3.2.10 Proposición (Asocitividad del grupo asociado a las curvas elípticas)**.** Dada curva elíptica E y fijado un punto base $\mathscr{O} \in E$, la operación $\oplus$ definida en el apartado anterior es asociativa.

Demostración. Probaremos la asociatividad en el caso de que todos los puntos que aparecen a continuación sean diferentes. En el caso de que suceda que la recta que corta a dos de ellos fuera tangente a

la curva, se seguiría un razonamiento análogo con rectas tangentes en vez de secantes. Sean $P,Q,R \in E$ distintos. Sea: $\begin{cases} L_1 \cap E = \{P,Q,S'\} \\ M_1 \cap E = \{\mathscr{O},S',S\} \\ L_2 \cap E = \{S,R,T'\} \\ L' \cap E = \{\mathscr{O},T',V'\} \end{cases}$ y sea: $\begin{cases} M_2 \cap E = \{Q,R,U'\} \\ L_3 \cap E = \{\mathscr{O},U',U\} \\ M_3 \cap E = \{P,U,T''\} \\ M' \cap E = \{\mathscr{O},T'',V''\} \end{cases}$

Como $(P \oplus Q) \oplus R = V'$ y $P \oplus (Q \oplus R) = V''$, donde V' es el punto de E formado con la intersección con E de la recta que pasa por O y T' y V'' es el punto de E formado con la intersección con E de la recta que pasa por O y T'', basta probar que $T' = T''$ y así ambas rectas(las que pasa por $\mathscr{O}$ y T' y la que pasa por $\mathscr{O}$ y T'') serán la misma; es decir, $V' = V''$.

Entonces ahora, buscamos aplicar el Corolario 3.2.9 a las curvas $\mathscr{C} = \mathscr{C}(F_1F_2F_3)$ y $\mathscr{D} = \mathscr{D}(G_1G_2G_3)$ donde $F_i, i = 1,2,3$ son las formas que definen las rectas L_i y $G_j, j = 1,2,3$ las formas que definen las rectas M_j. Así, obtenemos que:

$$E \cap \mathscr{C} = \{\mathscr{O},P,Q,R,S,S',U,U',T'\}, \tag{3.3}$$
$$E \cap \mathscr{D} = \{\mathscr{O},P,Q,R,S,S',U,U',T''\}. \tag{3.4}$$

donde vemos que como la cúbica $\mathscr{D}$ pasa por 8 de los 9 puntos de $E \cap \mathscr{C}$, por lo tanto por el Corolario 3.2.9 concluimos que este noveno punto es el mismo; es decir, $T' = T''$ como queríamos probar.

En las siguientes figuras podemos ver la construcción utilizada en esta demostración. Vemos que, efectivamente, los puntos T' y T'' coinciden.

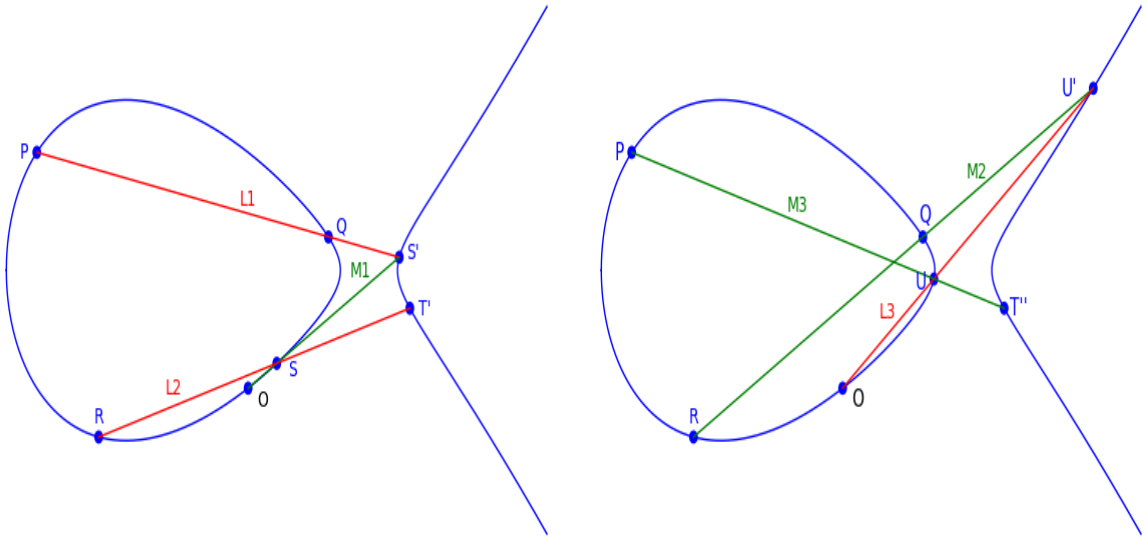

Figura 3.3: Ejemplo de asociatividad de P,Q y R en la curva elíptica $y^2 = x^3 - 2x + 1$ definida sobre $\mathbb{R}$. La figura de la izquierda muestra la operación $(P \oplus Q) \oplus R$ y la de la derecha $P \oplus (Q \oplus R)$.

□

Acabamos de demostrar la asociatividad de $\oplus$ sobre las curvas elípticas por lo que concluimos que si E es una curva elíptica, entonces $(E, \oplus)$ *es un grupo abeliano*.

3.3. Curvas elípticas sobre cuerpos finitos

A continuación vamos a estudiar el número de puntos racionales que tiene una curva elíptica. Para ello, tomamos para todo este apartado $\mathbb{F} = \mathbb{F}_q$ que representará un cuerpo finito de característica p. En la criptografía con curvas elípticas se suelen usar cuerpos finitos, de ahí la importancia estudiar el número de puntos de una curva elíptica definida sobre una curva elíptica.

3.3.1 Definición. Sea $E(\mathbb{F})$ una curva elíptica definida por su ecuación de Weierstrass. Decimos que $P=(P_1,P_2)$ es un punto racional de la curva si P es solución de dicha ecuación.

En primer lugar, notar que en general, una curva elíptica $E(\mathbb{F})$ definida sobre un cuerpo $\mathbb{F}$ no algebraicamente cerrado no tiene por qué tener puntos racionales (distintos de $\mathscr{O}$). Sin embargo, en este apartado vamos a probar que si $q>4$, entonces $E(\mathbb{F}_q)$ tiene al menos un punto racional no trivial.

Más aún, el objetivo es probar el *Teorema de Hasse*, el cual nos dará una estimación del número de puntos racionales de $E(\mathbb{F}_q)$. Supongamos pues para toda la sección $q>4$.

Para llegar a resultados más avanzados, es necesario explicar algunos resultados previos.

3.3.2 Definición. Sea $E(\mathbb{F})$ una curva elíptica y sea $\overline{\mathbb{F}}$ la clausura algebraica de $\mathbb{F}$. Un *endomorfismo* de $\overline{\mathbb{F}}$ es un homomorfismo $\alpha : E(\overline{\mathbb{F}}) \to E(\overline{\mathbb{F}})$ dado por funciones racionales (cocientes de polinomios). Dicho de otro modo, α preserva la suma y el elemento neutro de $E(\overline{\mathbb{F}})$.

El endomorfismo trivial que lleva cada punto a $\mathscr{O}$ lo denotaremos por o.

3.3.3 Lema. Si α es un endomorfismo de una curva elíptica definifida por su ecuación de Weierstrass (3.1), entonces α se puede escribir como:

$$\alpha(x,y)=(r_1(x),r_2(x)y),$$

donde r_1 y r_2 son funciones racionales. Entonces

- $r_1(x)=\frac{f(x)}{g(x)}$, con $f(x),g(x)$ polinomios sin factores comunes.
- Si $g(x)=0$ para algún punto (x,y), entonces definimos $\alpha(x,y)=\mathscr{O}$.
- Si $g(x)\neq 0$, entonces $r_2(x)$ está bien definida.

3.3.4 Definición. A continuación vamos a definir dos conceptos que vamos a necesitar:

- El grado de un endomorfismo α es $deg(\alpha)=max\{deg(f(x)),deg(g(x))\}$ si α es no trivial. Si lo es, $deg(o)=0$.
- Un endomorfismo α no trivial es *separable* si la derivada $r_1'(x)$ no es idénticamente cero.

El siguiente resultado será crucial en la demostración del Teorema de Hasse. Denotamos por $|C|$ al cardinal de un conjunto C.

3.3.5 Proposición. Sea $\alpha\neq 0$ un endomorfismo separable de una curva elíptica E. Entonces:

$$deg(\alpha)=|ker(\alpha)|.$$

donde $ker(\alpha)$ es el núcleo de del homomorfismo .

Demostración. Por el lema 3.3.3, $\alpha(x,y)=(r_1(x),r_2(x)y)$ con $r_1=\frac{f(x)}{g(x)}$. Supongamos que α es separable. Entonces $r_1'\neq 0$, por lo que $f'g-fg'$ no es el polinomio nulo. Sea S el conjunto de $x\in\overline{\mathbb{F}}$ tal que $(fg'-f'g)(x)=0$. Sea $(u,v)\neq(0,0)\in E(\overline{\mathbb{F}})$ cumpliendo:

1. $deg(f(x)-ug(x))=max\{deg(f),deg(g)\}=deg(\alpha)$.
2. $u\notin r_1(S)$.
3. $(u,v)\in\alpha(E(\overline{\mathbb{F}}))$.

Tenemos que ver que existen exactamente $deg(\alpha)$ puntos $(x_1,y_1) \in E(\overline{\mathbb{F}})$ tal que $\alpha(x_1,y_1) = (u,v)$. Para tales puntos, se tiene:

$$\frac{f(x_1)}{g(x_1)} = u \qquad\qquad y_1 r_2(x_1) = v$$

Como $(u,v) \neq \infty$, $g(x_1) \neq 0$ luego $r_2(x)$ está definido. Como $v \neq 0$ y $y_1 r_2(x_1) = v$, se tendrá $y_1 = \frac{v}{r_2(x_1)}$, esto es, x_1 determina y_1, por lo que solo tenemos que contar valores de x_1.

Por la propiedad (2) anterior, $f(x) - ug(x)$ tiene $deg(\alpha)$ raíces, contando multiplicidades. Veamos que $f - ug$ no tiene raíces múltiples. Supongamos que x_0 es una raíz múltiple. Entonces:

$$f(x_0) - ug(x_0) = 0 \qquad\qquad f'(x_0) - ug'(x_0) = 0$$

Multiplicando las ecuaciones $f = ug$ y $ug' = f'$ resulta

$$uf(x_0)g'(x_0) = uf'(x_0)g(x_0)$$

Como $u \neq 0$, esto implica que x_0 es una raíz de $fg' - f'g$, por lo que $x_0 \in S$. Así, $u = R_1(x_0) \in r_1(S)$, contrario a la propiedad (3) anterior. Concluimos con esto que $f - ug$ no tiene raíces múltiples y por ello tiene $deg(\alpha)$ raíces distintas. □

3.3.6 Lema. Sean $\alpha_1, \alpha_2, \alpha_3$ endomorfismos no triviales de una curva elíptica E con $\alpha_1 + \alpha_2 = \alpha_3$. Supongamos que cada endomorfismo está dado de la siguiente forma $\alpha_j(x,y) = (R_{\alpha_j}(x), yS_{\alpha_j}(x))$ y que existen constantes $c_{\alpha_1}, c_{\alpha_2}$ tales que:

$$\frac{R'_{\alpha_1}(x)}{S_{\alpha_1}(x)} = c_{\alpha_1} \qquad\qquad \frac{R'_{\alpha_2}(x)}{S_{\alpha_2}(x)} = c_{\alpha_2}$$

Entonces:

$$\frac{R'_{\alpha_3}(x)}{S_{\alpha_3}(x)} = c_{\alpha_1} + c_{\alpha_2}.$$

3.3.7 Proposición. Sea $E(\mathbb{F}_q)$ una curva elíptica. La aplicación definida por:

$$\phi_q(x,y) = (x^q, y^q) \qquad \phi_q(\mathcal{O}) = \mathcal{O}$$

es un endomorfismo y se llama *endomorfismo de Frobenius*.

Demostración. La demostración se sigue de manera inmediata teniendo en cuenta el *automorfismo de Frobenius*

$$\begin{aligned} \overline{\mathbb{F}}_q &\to \overline{\mathbb{F}}_q \\ x &\mapsto x^q \end{aligned}$$

□

Vamos a ver ahora propiedades de este endomorfismo que nos serán útiles más adelante.

3.3.8 Lema. Sea $E(\mathbb{F}_q)$ una curva elíptica y sea $(x,y) \in E(\overline{\mathbb{F}}_q)$. Entonces

$$(x,y) \in E(\mathbb{F}_q) \Leftrightarrow \phi_q(x,y) = (x,y)$$

Demostración. Usando $a^q = a, \forall a \in \mathbb{F}_q$, tenemos:

$$(x,y) \in E(\mathbb{F}_q) \Longleftrightarrow x,y \in \mathbb{F}_q \Longleftrightarrow \phi_q(x) = x \text{ y } \phi_q(y) = y \Longleftrightarrow \phi_q(x,y) = (x,y).$$

□

3.3.9 Proposición. Sea $E(\overline{\mathbb{F}}_q)$ una curva elíptica y consideremos el endomorfismo $\phi_q - 1$. Entonces:

1. $ker(\phi_q - 1) = E(\mathbb{F}_q)$.

2. $\phi_q - 1$ es separable, por lo que $|E(\mathbb{F}_q)| = deg(\phi_q - 1)$.

Demostración. (1) es consecuencia del lema 3.3.8 y la definición 3.3.4. Para ver(2), tenemos que ver que $\phi_q - 1$ es separable.

Para ello, por 3.3.6 tenemos que

$$\frac{R'_{\phi_q}(x)}{S_{\phi_q}(x)} = 0 \qquad\qquad \frac{R'_{-1}(x)}{S_{-1}(x)} = 1$$

por lo tanto

$$\frac{R'_{\phi_q - 1}(x)}{S_{\phi_q - 1}(x)} = c_{\phi_q} + c_{-1} = 1,$$

luego $R'_{\phi_q - 1}(x) \neq 0$, lo cual implica que $\phi_q - 1$ es separable y la segunda parte se sigue aplicando la proposición3.3.5. □

3.3.10 Lema. Sean α y β endomorfismos de E y sean r, s enteros. Entonces $deg(r\alpha + s\beta) = r^2 deg(\alpha) + s^2 deg(\beta) + rs(deg(\alpha + \beta) - deg(\alpha) - deg(\beta))$.

Así, para el endomorfismo definido en la proposición anterior, tenemos que $deg(r\phi_q) - s) = r^2 q + s^2 - rsa$, donde $a = deg(\phi_q) + deg(-1) - deg(\phi_q - 1)$.

Demostración. La primera parte no la vamos a demostrar por cuestión de espacio. Veamos la segunda parte como consecuencia directa de la primera:

$$deg(r\phi_q - s) = r^2 deg(\phi_q) + s^2 deg(-1) + rs(\underbrace{deg(\phi_q - 1) - deg(\phi_q) - deg(-1)}_{-a})$$

Como $deg(\phi_q) = q$ y $deg(-1) = 1$, el resultado se deduce de 3.3.9. □

Ahora ya estamos en disposición de enunciar y demostrar el teorema final:

3.3.11 Teorema (Teorema de Hasse)**.** Sea $E(\mathbb{F}_q)$ una curva elíptica. Entonces el orden de $E(\mathbb{F}_q)$ verifica

$$| q + 1 - |E(\mathbb{F}_q)| | \leq 2\sqrt{q}. \tag{3.5}$$

Demostración. Denotamos a como en el lema 3.3.10 y obtenemos:

$$|a| = |deg(\phi_q) + deg(-1) - deg(\phi_q - 1)| = |q + 1 - |E(\mathbb{F}_q)||$$

donde estamos usando la proposición 3.3.9. Veamos que $|a| \leq 2\sqrt{q}$.

Como $deg(r\phi_q - s) \geq 0$, el lema anterior 3.3.10 implica que

$$q\left(\frac{r}{s}\right)^2 - a\left(\frac{r}{s}\right) + 1 \geq 0,$$

para cualesquiera enteros r, s. Como el conjunto de los racionales es *denso* en $\mathbb{R}$, tenemos que

$$qx^2 - ax + 1 \geq 0,$$

para todo número real x. Así, el discriminante de este polinomio es negativo o cero, lo que implica que $a^2 - 4q \leq 0$, luego $|a| \leq 2\sqrt{q}$. □

Así, gracias al Teorema de Hasse, podemos obtener un rango para el número de puntos que posee una curva elíptica definida sobre un cuerpo finito. Para concluir, habíamos comentado al inicio de la sección que si el cuerpo $\mathbb{F}$ tenía característica mayor que 4, entonces la curva posee algún punto racional aparte del trivial. Ahora vemos que si $q = 5$, entonces $|E(\mathbb{F}_q)| \in [q+1-2\sqrt{q}, q+1+2\sqrt{q}] = [6-2\sqrt{5}, 6+2\sqrt{5}] \approx [1{,}53, 10{,}47]$, luego $E(\mathbb{F}_5)$ poseerá por lo menos 2 puntos racionales y a lo sumo 10.

El Teorema de Hasse nos proporciona una cota bastante ajustada de la cardinalidad de una curva elíptica definida sobre un cuerpo finito. Cuando q es grande: $|E(\mathbb{F}_q)| \in [q-2\sqrt{q}, q+2\sqrt{q}]$ donde $\sqrt{q}$ es mucho más pequeño que q. Veamos esto en un ejemplo:

3.3.12 Ejemplo. [1] Tomamos $q = 340282366920938463481821351505477763161$ y la curva elíptica $E(\mathbb{Z}_q) := y^2 = x^3 + 4x - 3$.
La cardinalidad de esta curva es:$|E(\mathbb{Z}_q)| = 340282366920938463512281837415684970070$ que como observamos comparte los primeros dígitos "34028236692093846" y difiere en el resto. Ya hemos probado que algo similar va a suceder siempre, ya que la cardinalidad de la curva puede diferir a lo sumo en $2\sqrt{q}$ de q. En este caso,

$$|q - |E(\mathbb{Z}_q)|| = 30460485910207206909 < 36893488147419103234 \approx 2\sqrt{q}$$

[1] Ejemplo realizado con SAGE. Para más información consultar [13]

Capítulo 4

Criptografía sobre curvas elípticas

En esta sección vamos a ver las diferentes aplicaciones y ventajas de las curvas elípticas en la criptografía. Trataremos el problema del logaritmo discreto con curvas elípticas y los principales modelos criptográficos implementados con curvas elípticas, haciendo mención especial a los esquemas de firma digital basados en curvas elípticas.

En los esquemas criptográficos de clave pública, la pareja de claves se selecciona de tal modo que el problema de calcular la clave privada a partir de la clave púbica sea equivalente a resolver un problema computacional intratable. Los dos problemas más usados en los sistemas de clave pública son: El problema de factorización de enteros (usado en el protocolo RSA) y el problema del logaritmo discreto.

Dentro del problema del logaritmo discreto, nos vamos a centrar en el logaritmo discreto con curvas elípticas, ya que al utilizar el grupo de una curva elíptica sobre un cuerpo finito $(E(\mathbb{F}_q))$, se aumentará la eficiencia computacional debido a que no necesitamos claves tan largas para obtener la misma seguridad. Esto se muestra en la tabla 1, donde se compara el nivel de bits necesarios para obtener el mismo nivel de seguridad entre el protocolo Diffie-Hellman (DH) sobre un cuerpo finito y el mismo protocolo usando el grupo de una curva elíptica sobre el mismo cuerpo finito (ECDH), que explicaremos más adelante.

Nivel de seguridad(bits)	Tamaño de clave DH(bits)	Tamaño de clave en ECDH (bits)
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	521

Tabla 1: Tamaños de claves recomendados por NIST [1].

Las ventajas en el uso de parámetros más pequeños incluyen velocidad, claves y certificados más pequeños.

Problema del logaritmo discreto

Siguiendo lo que acabamos de introducir, la dificultad del problema del logaritmo discreto es la base para la seguridad de la criptografía basada en curvas elípticas.

4.0.1 Definición. El *problema del logaritmo discreto sobre curvas elípticas (ECDLP)* consiste en lo siguiente: dada una curva elíptica E definida sobre un cuerpo finito $\mathbb{F}_q$, un punto $G \in E(\mathbb{F}_q)$ de orden n y un punto $P \in \langle G \rangle$, encontrar el entero $k \in [0, n-1]$ tal que $P = kG$. El entero k se llama el *logaritmo discreto de P respecto de la base G*.

De aquí en adelante consideraremos $E(\mathbb{F}_q)$ una curva elíptica en su forma de Weiertrass($y^2 = x^3 + ax + b$) y $\mathbb{F}_q$ un cuerpo finito.

4.0.2 Protocolo (*Generación de claves*)**.** Sea $D = (E(\mathbb{F}_q), \mathbb{F}_q, G, n)$ los *parámetros de dominio*, donde $E(\mathbb{F}_q)$ es una curva elíptica definida sobre el cuerpo $\mathbb{F}_q$, $G \in E(\mathbb{F}_q)$ y n su orden ($nG = \mathcal{O}$), que tendrá que ser grande para que sea seguro. Los pasos para generar claves son:

1. Seleccionar $d \in [2, n-2]$ aleatoriamente. Descartamos los casos $d = 1, n-1$ ya que en estos casos los protocolos basados en el *PLD* pierde toda la seguridad.
2. Calcular $P = dG$.
3. Devolver (P, d) donde P es la clave pública y d la privada.

4.0.3 Comentario. Se suele tomar G un generador del grupo de la curva en el caso de que este grupo sea cíclico. Lo habitual es trabajar con curvas preestablecidas de las cuales se conoce un generador. Este es el caso por ejemplo de la curva $secp256k1$ [10].

4.0.4 Nota. La forma de calcular dG se realiza calculando

$$\begin{aligned} G+G &= 2G \\ 2G+2G &= 4G \\ &\vdots \\ 2^iG+2^iG &= 2^{i+1}G \end{aligned}$$

Cualquier dG puede ser calculado mediante la suma de algunos de estos puntos de la forma 2^jG. Notar además que dada una curva elíptica solamente necesitamos calcular estos puntos una vez y por tanto tan solo tenemos que operar con ellos para obtener cualquier dG. Así, por ejemplo para hallar $P = 2^{256}G$ bastaría con hacer 256 operaciones, mientras que si lo quisiéramos realizar sumando directamente, tendríamos que realizar 2^{256} lo cual sería computacionalmente imposible.

Además, si fuera posible calcular estas operaciones, obtener la clave privada d nos costaría lo mismo que calcular la clave pública $P = dG$, ya que:
$P - G = (d-1)G, P - G - G = (d-2)G, \ldots$ y así sucesivamente restar G a lo que se vaya obteniendo hasta obtener G, con lo que conseguiríamos la clave privada d, que será el número de veces que se resta G.

4.1. Protocolos criptográficos

Vamos a ver ahora varios ejemplos sobre protocolos clásicos de criptografía asimétrica basados en el problema del logaritmo discreto sobre curvas elípticas.

4.1.1 Protocolo (*Diffie-Hellman(ECDH)*)**.** El siguiente procedimiento permite a dos personas Alicia(A) y Bob(B) acordar de forma segura el valor de un punto en una curva elíptica, de forma que aunque alguien observe los mensajes intercambiados entre ellos, no va a poder hallar el punto acordado.

1. A y B acuerdan unos parámetros de dominio $D = (E(\mathbb{F}_q), \mathbb{F}_q, G, n)$.
2. A calcula su pareja de claves (P_A, d_A) según 4.0.2. B hace lo mismo calculando (P_B, d_B).
3. A y B intercambian sus claves públicas P_A, P_B.
4. A calcula d_AP_B y B calcula d_BP_A. Ambos calculos devuelven el punto $(d_Ad_B)G$.

La utilidad de este protocolo reside en permitir que A y B acuerden una clave por un canal inseguro que pueda ser utilizada posteriormente para transmitir datos sobre un esquema de cifrado simétrico como el *AES* (*Advanced Encryption Standard*). Esta clave se suele obtener extrayendo unos cuantos

bits de la coordenada x de $(d_A d_B)P$ o evaluando una *función hash* en dicha coordenada. Está claro que otra persona en el sistema conoce los parámetros de dominio y las claves públicas P_A y P_B, por lo que si fuera capaz de resolver el *ECDLP*, podría encontrar las claves privadas d_A, d_B y recuperar el punto secreto $(d_A d_B)G$.

4.1.2 Nota (Funciones hash criptográficas)**.** Las funciones hash criptográficas son aquellas que, dada una entrada, nos proporcionan una salida de tamaño fijo y fácil de calcular. Veremos que estas funciones son muy usadas en los distintos protocolos, debido a unas ciertas propiedades que son exigidas a estas funciones:

- Dada una x, que sea sencillo calcular su imagen $h(x)$ para que sean utilizables en la práctica, pero que, sin embargo, teniendo $h(m)$ sea muy difícil hallar una preimagen.
- Se pide que sea difícil encontrar una pareja $(x,y), x \neq y$ tal que $h(x) = h(y)$ (resistencia a colisiones).
- Se pide que sean *uniformes*; es decir, que una muestra de valores que son imágenes de una función hash no sea diferenciable de una muestra aleatoria simple.

Algunas de estas funciones más utilizadas y recomendadas por el NIST son $SHA-256, SHA-512$ debido a sus propiedades y seguridad. Para más información, se recomienda consultar [7, 8].

4.1.3 Protocolo (*ElGamal*)**.** Dados unos parámetros de dominio $D = (E(\mathbb{F}_q), \mathbb{F}_q, G, n)$. A quiere mandar un mensaje m cifrado a B. Entonces:

- B escoge un entero d_B que será su clave privada y calcula con él su clave pública P_B, la cual envía a A.
- A escoge ahora al azar un número $k \in \mathbb{F}_p$ y calcula $P = kG$.
- A calcula P_m el punto de la curva asociado al mensaje m.
- Cifra P_m como $C = P_m + kP_B$, y envía a B (C, P).

Para descifrar el mensaje, B deberá calcular $P_m = C - d_B P = C - d_B kG = C - kP_B$ y encontrar el mensaje asociado a P_m. Vemos que la dificultad en este caso se encuentra en hallar k y d_B.

4.2. Esquemas de firma digital con curvas elípticas

En esta sección nos vamos a dedicar al estudio de varios esquemas de firma digital basadas en curvas elípticas. En el año 1991 el *NIST* (*National Institute of Standards and Technology*) propuso el *DSS* (*Digital Signature Standard*) basado en el *DSA* (*Digital Signature Algorithm*) como estándar de firma digital. El *DSS* está basado en el protocolo *ElGamal*. Sin embargo, vamos a ver el protocolo *ECDSA* (*Elliptic Curve Digital Signature Algorithm*), puesto que se ha convertido en el esquema estándar de firma digital con curvas elípticas, usado por ejemplo en Whatsapp y Bitcoin. Después pasaremos a dar los *esquemas de firma Schnorr* de las que resaltaremos sus buenas propiedades.

La finalidad de las firmas digitales es la siguiente: imaginemos que A quiere enviar a otra persona, por ejemplo a B, un mensaje firmado (m). Una vez B los reciba, tiene que comprobar que el mensaje ha sido firmado por A y no por otra persona. Para ello, lo primero que se hace es crear una clave pública y una privada a partir de una curva elíptica y un punto base de la misma, prefijados y públicos y después se realiza el proceso de firma con su correspondiente verificación.

4.2.1 Esquema (*ECDSA*)**.** El siguiente procedimiento permite a A firmar un documento digital m y a B verificar que la firma es válida.

1. A y B acuerdan unos parámetros del dominio $D = (E(\mathbb{F}_q), \mathbb{F}_q, G, n)$.

2. A calcula su pareja de claves (P,d) pública y privada respectivamente, escogiendo d un número al azar entre 2 y $n-2$ y calculando $dG = P$.

3. A calcula $h(m)$, donde $h(\cdot)$ es una función hash.

4. A calcula la firma (r,s) de $h(m)$ como sigue:

   - Selecciona $k \in [2, n-2]$ aleatoriamente y calcula el punto $kG = (x,y)$. A k se le denomina clave privada efímera.
   - Calcula $r \equiv x \pmod{n}$.
   - Calcula $s \equiv k^{-1}(h(m) + dr) \pmod{n}$.

5. A publica la firma (r,s) y $(h(m),P)$.

6. B verifica la firma de la siguiente forma:

   - Calcula $u_1 \equiv h(m)s^{-1} \pmod{n}$ y $u_2 \equiv rs^{-1} \pmod{n}$.
   - Calcula el punto $(x_0,y_0) = u_1G + u_2P$ y $v \equiv x_0 \pmod{n}$.
   - Comprueba si $v \equiv r \pmod{n}$. En caso afirmativo la firma es válida. En caso negativo, la firma no es válida.

Demostración de que la verificación de la firma funciona: Si una firma (r,s) de un mensaje m fue generada por el signatario legítimo, entonces $s \equiv k^{-1}(h(m) + dr) \pmod{n}$. Reordenando se tiene

$$k \equiv s^{-1}(h(m) + dr) \equiv s^{-1}h(m) + s^{-1}rd \equiv u_1 + u_2 d \pmod{n}$$

Así, $(x_0,y_0) = u_1G + u_2P = (u_1 + u_2 d)G = kG$, por lo que $v = r$ como se requería.

4.2.2 Comentario. (*Seguridad ECDSA*)

- Análogamente al protocolo *ECDH*, si un atacante pudiese calcular logaritmos discretos, podría calcular la clave privada d_A y firmar como A cada mensaje.

- Si el número aleatorio k se reutiliza, un atacante podría recuperar d sin necesidad de resolver el *ECDLP*. Esto se debe a lo siguiente:

  - Si k se repite en dos firmas, entonces es claro que r es el mismo.
  - De la expresión $s = k^{-1}(h(m) + dr)$ podemos despejar k.
  - Tenemos entonces $k = s_1^{-1}(h_1(m) + dr)$ en la primera firma y $k = s_2^{-1}(h_2(m) + dr)$ de la segunda, por lo que podemos igualar estas expresiones y extraer d en función de $s_1, s_2, h_1(m)$, $h_2(m), r$ ya que se conoce todo.

  Por esto, k debe ser generado de forma segura para cada mensaje, almacenado de forma segura y destruido después de haber sido utilizado.

- La generación de la clave también es un proceso a estudiar. La mayoría de ataques al protocolo ECDSA suelen tener relación con esto, como es el caso de los ataques basados en el *algoritmo LLL*(consultar [9]), el cual funciona cuando la clave efímera k está generada bajo ciertas condiciones o cuando una pequeña fracción de ella se ha revelado. Sin embargo, estos ataques no son eficientes cuando no se conoce nada de k por lo que esta debe ser escogida de forma segura. Podemos pensar en lo siguiente: si un atacante aprende a hallar k puede recuperar d sistemáticamente. Sin embargo, varios resultados, que se pueden consultar en [9] demuestran que resolver el problema de hallar d es computacionalmente igual de complicado que resolver el problema del logaritmo discreto.

4.2.1. Firmas Schnorr

Ahora vamos a dar las nociones básicas principales sobre otro esquema de firma digital, el *esquema de firma Schnorr*. Este esquema fue descrito y patentado por Claus Schnorr, quien mantuvo la patente sobre él hasta febrero del año 2008. Este tipo de firmas es conocido por su simplicidad, lo cual le proporciona ciertas propiedades que lo dotan de una gran utilidad. La seguridad de las firmas Schnorr se basa también en el problema del logaritmo discreto. Vamos a ver el proceso firma y comprobación de la misma:

4.2.3 Esquema (*Firma Schnorr*)**.**

- A y B acuerdan unos parámetros de dominio $D = (E(\mathbb{F}_q), \mathbb{F}_q, G, n)$.
- A escoge al azar su clave privada k y calcula $P = kG$, donde P será su clave pública.
- A escoge otro número al azar r que será su clave efímera y calcula $R = rG$, que será su clave pública efímera.
- A envía a B nuestro mensaje (m), P y R.
- La firma actual es creada al calcular una función Hash(H) a toda la información pública para crear $e = H(R\|P\|m)$. Ahora la firma es construida usando nuestra información privada: $s = r + k \cdot e$.

B puede calcular e, ya que sabe m, P, R, pero no puede saber nuestras claves privadas. Vemos lo siguiente:

$$\begin{aligned} sG &= (r + k \cdot e)G \\ sG &= rG + e(kG) \\ sG &= R + eP \end{aligned}$$

Así, B debe calcular la clave pública correspondiente a la firma sG y comprobar que esta es igual a $R + eP$, de lo que conoce todo.

Pero, ¿por qué necesitamos ambas claves P y R? Supongamos que solo tenemos $P = kG$. Ahora $e = H(P\|m) \Rightarrow s = ek$. Igual que antes, podemos verificar que la firma es válida del mismo modo: $sG = ekG = eP$. Pero el problema ahora está en que cualquiera puede calcular nuestra clave privada porque al ser s un número, entonces $k = \frac{s}{e}$, lo cual no es complicado de calcular. Sin embargo, al añadir la clave pública efímera $R = rG$, podemos solucionar este problema ya que entonces $k = \frac{s-r}{e}$, donde r es desconocida.

Ventajas de los esquemas de firma Schnorr: La principal ventaja que podemos observar es la linealidad. Esto puede ser usado como vamos a ver a continuación para crear *esquemas de firma múltiple* y *esquemas de agregación de firmas*.

Esquemas de agregación de firmas y esquemas de firma múltiple

El término *agregación de firmas* hace referencia a las firmas múltiples que aparecen como una firma de longitud sublineal en el número real número real de signatarios. Por ejemplo, pueden existir 10 signatarios y que figure como una única firma. Así, quien tiene que verificar las firmas no necesita del conocimiento de las claves públicas originales de los participantes, pudiendo recibir únicamente la clave agregada. Esto proporciona una mayor compactificación de la información al mismo tiempo que aumenta la privacidad de los signatarios. Por ejemplo *MuSig* es un esquema de agregación de firmas para las firmas Schnorr.

Veamos cómo la propiedad de la linealidad de las firmas Schnorr puede ser usada para construir un esquema de firma múltiple.

Imaginemos que A y B quieren firmar conjuntamente un mensaje sin tener que confiar en el otro. Entonces necesitan ser capaces de demostrar la autoría de sus respectivas claves. Además, la firma agregada será válida solamente si ambos, A y B, aportan su parte de la firma.

Para realizar el primer acercamiento a los esquemas de firma múltiple, un esquema de firma múltiple debe utilizar firmas agregadas sublineales siguiendo las siguientes propiedades:

- Debe satisfacer la ecuación normal de Schnorr, donde las firmas resultantes pueden ser escritas como una combinación de las claves públicas.
- Debe admitir que todos los signatarios sean requeridos para cooperar.
- Debe admitir que la agregación pueda ser hecha por cualquiera.
- Debe admitir que cada signatario firme el mismo mensaje
- Debe admitir que cada signatario firme su propio mensaje.

4.2.4 Esquema (*Firma múltiple*)**.** El esquema de firma múltiple *MuSig*, se trata de un esquema que verifica todas las premisas anteriores. Este esquema permite que varios signatarios firmen el mismo mensaje, necesitándose la colaboración de todos ellos. Esto se consigue de la siguiente manera:

1. Todos los signatarios conocen los parámetros de dominio $D = (E(\mathbb{F}_q), \mathbb{F}_q, G, n)$.
2. Cada signatario tiene una clave pública y privada, $P_i = k_i G$ y una clave pública y privada efímera $R_i = r_i G$ como antes.
3. Cada signatario comparte con el resto de signatarios su clave pública efímera, R_i.
4. Todos los signatarios calculan la clave pública compartida, P, como sigue:

$$\begin{aligned} l &= H(P_1 \| \cdots \| P_n) \\ a_i &= H(l \| P_i) \\ P &= \sum a_i P_i \end{aligned}$$

5. Todo signatario calcula también la clave pública efímera compartida, $R = \sum R_i$.
6. e ahora es $H(R\|P\|m)$.
7. Cada signatario aporta su contribución a la firma como:

$$s_i = r_i + e k_i a_i.$$

Notar que la única diferencia aquí de una firma Schnorr estándar es la inclusión del factor a_i. La firma agregada es la suma, $s = \sum s_i$. La verificación se realiza como es habitual:

$$sG \equiv R + eP$$

Demostración.

$$sG = \sum s_i G = \sum (r_i + k_i a_i e) G = \sum r_i G + \sum k_i G a_i e = \sum R_i + e \sum a_i P_i = R + eP$$

□

4.2.5 Comentario. Notar que los esquemas de firma múltiple solamente se pueden usar con firmas Schnorr debido a su linealidad y que por ejemplo, no se puede implementar en *ECDSA*, ya que este esquema de firma no es lineal. Esta es la principal desventaja que posee *ECDSA*, que lo hace menos útil frente a los esquemas de firma Schnorr que venimos recalcando en esta sección.

4.3. Implementación de las curvas elípticas en la actualidad

Una vez ya hemos visto los protocolos y esquemas de firmas más usados con curvas elípticas, vamos a explicar algunas aplicaciones de las curvas elípticas en la criptografía en el presente.

Entre las funciones de las curvas elípticas hoy en día, destaca su uso para realizar transacciones con criptomonedas. Entre las criptomonedas más utilizadas en la actualidad, que además utiliza curvas elípticas (ECDSA) en sus transacciones, destaca *Bitcoin*. Una de las características de esta criptomoneda es que almacena toda la información de sus transacciones. Esto supone un inconveniente por todo el espacio de almacenamiento que necesita, lo que implica que una persona que se incorpore al sistema y desee verificar la validez de una moneda, debe descargar y revisar todas las transacciones que se han realizado desde su creación hasta la actualidad. Para buscar una solución a esto se creó *Mimblewimble*.

Mimblewimble es un protocolo creado a mediados de 2016 que solamente utiliza firmas para realizar las transacciones. Su objetivo fue crear un nuevo esquema de funcionamiento centrado en la privacidad, al mismo tiempo que se ahorrara espacio de almacenamiento.

Una moneda en Mimblewimble C es de la siguiente manera:

$$C = xG + rH,$$

donde x es el *valor* de la moneda y r es un número escogido al azar que será el factor de ocultamiento, conocida por el propietario de la moneda; G y H son puntos de una curva elíptica $E(\mathbb{F}_q)$ tales que no se puede calcular el logaritmo discreto de uno respecto del otro.

Una transacción en Mimblewimble consiste en gastar unas monedas, a las que llamaremos *entradas*, a cambio de generar otras monedas a las que llamaremos *salidas*.

El principio básico que todas las transacciones deben cumplir es que la suma de los valores de las entradas sea igual a la suma de los valores de las salidas. Ilustremos esto con un ejemplo.

4.3.1 Ejemplo. Imaginemos que existen dos personas A y B. Han pactado que A le venda a B un cierto valor x_3, entonces:

1. A posee dos monedas $C_1 = x_1G + r_1H, C_2 = x_2G + r_2H$ tales que $x_1 + x_2 \geq x_3$.
2. A calcula el valor $x_4 = x_1 + x_2 - x_3$, para que se cumpla $x_1 + x_2 = x_3 + x_4$.
3. A escoge al azar r_4 para generar una nueva salida $C_4 = x_4G + r_4H$.
4. A le envía a B C_1, C_2, C_4 y $r_1 + r_2 - r_4$.
5. B escoge al azar r_3' y calcula $r_3 = r_1 + r_2 - r_4 - r_3'$.
6. B genera una salida $C_3 = x_3G + r_3H$.
7. B genera la transacción $C_1 + C_2 - C_3 - C_4$ y se la envía a A.
8. B firma la transacción con el par de claves $(r_3', r_3'H)$ ya que:

$$C_1 + C_2 - C_3 - C_4 = \underbrace{(x_1 + x_2 - x_3 - x_4)}_{0}G + (r_1 + r_2 - r_4 - r_3)H = r_3'H,$$

   y le comunica esta firma a A.

4.3.2 Nota.

- A sabe que la firma ha sido generada por B porque nadie más conoce r_3'.
- B no puede generar C_3 con otro valor que no sea el acordado x_3, ya que sino en la transacción que genera no se anularía el factor correspondiente al punto G y necesitaría hallar el logaritmo discreto de un punto respecto del otro, lo cual hemos comentado que no puede realizarse.

- En este ejemplo se han gastado las monedas C_1 y C_2 y se han generado C_3 que pertenecerá a B y C_4 que pertenecerá a A.

Observamos que en cada transacción se anulan las partes correspondientes a los valores. Esto implica que, si queremos verificar la validez de una moneda, tan solo necesitaremos saber sus características actuales (valor y factor de ocultamiento) y las iniciales, ya que, si queremos verificar una moneda, la suma de todas las transacciones intermedias hará que estas se anulen entre sí y solamente nos quedarán el momento actual y el origen de la moneda.

Este hecho es el que nos permite compactificar toda la información en Mimblewimble a la vez que se aumenta la privacidad al no necesitarse el almacenamiento de las transacciones anteriores al momento actual de la moneda.

Ya se ha comentado el hecho de que Mimblewimble solamente utiliza firmas. Esto conlleva que Mimblewimble no soporte scripts como sí lo hacen otras criptomonedas. Muchas de ellas soportan incluir condiciones complejas que deben incluirse para que pueda gastarse una moneda, esto recibe el nombre de *contrato inteligente*. Es esta la razón por la que utiliza otros protocolos para llevar a cabo estos *contratos inteligentes*, como son las *Scriptless Scripts*.

Las Scriptless Scripts buscan añadir condiciones a los contratos inteligentes de Mimblewimble, sin modificar el sistema explicado antes, para que si dos partes consiguen ejecutar algún tipo de transacción fielmente; es decir, sin engañar al otro, entonces generarán una firma válida sin la necesidad de almacenar ningún detalle más del contrato original que la propia firma. Este se trata de uno de los motivos por lo que se usan Scriptless Scripts, la privacidad que proporciona el hecho de que solamente se almacenen las firmas de las transacciones. El hecho de que solamente se usen firmas puede parecernos a priori bastante restrictivo, pero vamos a ver a continuación que solamente con el uso de firmas poseemos lo necesario para realizar ciertos tipos de contratos inteligentes en Mimblewimble.

4.3.3 Nota. Notar que el esquema de firma múltiple Schnorr es ya una Scriptless Script, puesto que necesita la firma de cada uno de los signatarios $\{s_i\}$, pero se valida y se almacena tan solo la suma de ellas s, sin importar ningún detalle más, como por ejemplo el número signatarios que han firmado. Este esquema de firma se puede modificar para crear un nuevo esquema de firma digital como son las *firmas adaptadoras*.

Firmas adaptadoras

El objetivo de las firmas adaptadoras es llevar a cabo una transacción entre dos partes sin la necesidad de que una parte confíe en la otra, bajo la condición de que en el momento en el que una parte firme la transacción para obtener lo acordado, esta revele a la otra parte un secreto que podrá usar para obtener su parte del acuerdo.

Las firmas adaptadoras, entre otras funciones, permiten crear firmas para las transacciones con monedas en Mimblewimble, para que una transacción solamente pueda completarse si una parte revela un secreto a la otra con su firma.

Veamos ahora más detalladamente cómo A puede proporcionar a B una firma adaptadora. En esencia lo que va a suceder es que en el momento de firmar, A le va a revelar a B el logaritmo discreto de un punto, que será el secreto que hemos comentado.

4.3.4 Esquema (*Firma adaptadora*)**.**

1. A escoge al azar t y calcula $T = tG$.
2. Considerar el esquema de firma múltiple Schnorr 4.2.4 entre A y B, modificado de tal forma que A en vez de proporcionar su clave pública efímera R_A, le transmite $R_A + T$ a B.
3. A también envía a B T junto a su firma s_A.
4. Al conjunto $(T, T + R_A, s_A)$ es a lo que llamamos *firma adaptadora*.

5. B antes de firmar, verifica $s_A G = R_A + H(P\|R+T\|m)P_A$ y si esta firma es correcta, entonces B le envía su firma s_B .

6. La firma final $s = s_A + s_B$ es verificable pero no es válida. La firma $s+t$ sí que es válida.

7. Teniendo ya A la firma de B, entonces firma con la firma válida $s+t$.

8. En ese preciso momento, B conocerá s (ya la conocía desde antes) y $+t$ que le acaba de ser revelada, por lo que podrá calcular t, obteniendo así B el secreto.

4.3.5 Comentario.

- Notar que el paso 8 se realiza de forma inmediata después del 7. Luego, A firmará la transacción si y sólo si le revela a B su secreto. No puede suceder una cosa sin que suceda la otra. Así, si A decide detenerse en el momento justo antes de firmar, no puede firmar la transacción ni por tanto obtener nada de B.

- Se ha comentado en el protocolo anterior que la firma (s,R) no es válida. Esto se debe a que el punto que hemos usado para la función Hash es $R+T$ no R. Entonces B no puede obtener una firma válida. Para ello, necesitaría resolver el *ECDLP*. Lo que sucede es que después de la validación de la firma adaptadora, B recibe la firma valida $(s+t,R)$ lo cual le va a revelar el secreto t buscado.

- Notar también que cualquier observador de la cadena de bloques no distingue una firma común de una firma adaptadora y solamente verá el proceso cuando la firma esté finalmente validada y la transacción añadida ya a la cadena de bloques.

Bibliografía

[1] E. BARKER, W. BARKER, W. BURR, W. POLK Y M. SMID, *Recommendation for Key Management – Part 1: General (Revised)*, NIST Special Publication 800-57,Part 1 Revised, 2007, disponible en https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-57p1r2007.pdf.

[2] A.GIBSON, *Flipping the scriptless script on Schnorr*, 2018, https://joinmarket.me/blog/blog/flipping-the-scriptless-script-on-schnorr/

[3] D. GÓMEZ, *Curvas algebraicas*, pp 16–22, 2012, lhttp://www.mat.ucm.es/~arrondo/calg.pdf

[4] A. HOMERO, *Curvas elípticas en criptografía*, 2016, https://www.ugr.es/~anillos/textos/pdf/2016/CurvasElipticas.pdf

[5] J. MALDONADO, *¿Qué es Mimblewimble?*, 2021, https://es.cointelegraph.com/explained/what-is-mimblewimble

[6] G. MAXWELL, A. POELSTRA, Y. SEURIN Y P. WUILLE, *Simple Schnorr Multi-Signatures with Applications to Bitcoin*, ANSSI, París, Francia, 2018, https://eprint.iacr.org/2018/068.pdf

[7] NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY, *Secure Hash Standard (SHS)*, FIPS PUB 180-4, 2015, disponible en https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf.

[8] NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY, *SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions*, FIPS PUB 202, 2015, disponible en https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.202.pdf.

[9] PHONG Q. NGUYEN Y IGOR E. SHPARLINSKI, *The insecurity of the Elliptic Curve Digital Signature Algorithm with Partially Known Nonces*, DOI:10.1023/A:1025436905711, 2001, https://www.researchgate.net/profile/Igor-Shparlinski/publication/2406712_The_Insecurity_of_the_Elliptic_Curve_Digital_Signature_Algorithm_with_Partially_Known_Nonces/links/0912f50d93f7ac2275000000/The-Insecurity-of-the-Elliptic-Curve-Digital-Signature-Algorithm-with\-Partially-Known-Nonces.pdf

[10] M.M. PAYERAS, A.P. ISERN, M. MUT, *Sistemas de Pago Electrónico.*, Lección 3. Introducción a Bitcoin, 2014 http://www.criptored.upm.es/crypt4you/temas/sistemaspago/leccion3/leccion03.html#apartado32.

[11] A. POELSTRA, *Scriptless Scripts*, 2017, https://download.wpsoftware.net/bitcoin/wizardry/mw-slides/2017-03-mit-bitcoin-expo/slides.pdf

[12] A. POELSTRA, *MimbleWimble and Scriptless Scripts*, 2018, https://www.youtube.com/watch?v=ovCBT1gyk9c.

[13] THE SAGE DEVELOPMENT TEAM, *Sage 9.3 Reference Manual: Elliptic curves*, https://doc.sagemath.org/html/en/reference/arithmetic_curves/sage/schemes/elliptic_curves/ell_point.html.

[14] C. SHARROCK, SW. VAN HEERDEN, H. ODENDAAL, Y. ROODT Y D. ANSELL , *Introduction to Schnorr Signatures*, https://tlu.tarilabs.com/cryptography/digital_signatures/introduction_schnorr_signatures.html

[15] P. WUILLE, *Key Aggregation for Schnorr Signatures*, 2018, https://blockstream.com/2018/01/23/en-musig-key-aggregation-schnorr-signatures/