

# Elliptic Curve Ccriptography

Paolo Bettelini

## Contents

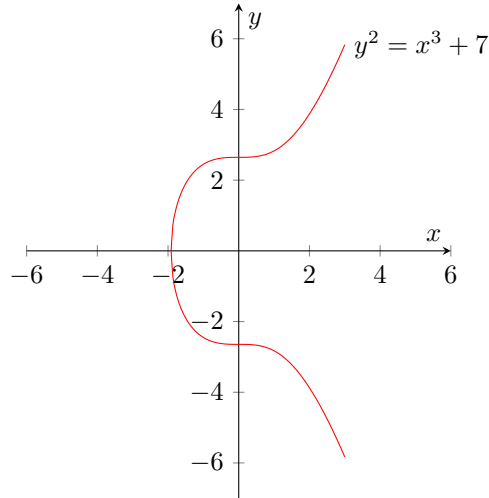
<b>1</b>	<b>Elliptic Curves</b>	<b>2</b>
1.1	Definition . . . . .	2
1.2	Addition . . . . .	2
1.3	Scalar Multiplication . . . . .	3
<b>2</b>	<b>Discrete logarithm problem</b>	<b>4</b>
2.1	Definition . . . . .	4
2.2	Diffie–Hellman . . . . .	4

# 1 Elliptic Curves

## 1.1 Definition

An elliptic curve  $E$  is a set of points such that

$$E = \{(x, y) \mid y^2 = x^3 + ax + b\} \cup \{O\}, \quad 4a^3 + 27b^2 \neq 0$$



Where  $O$  is a point at infinity.

The elliptic curve is symmetrical about the x-axis.

The opposite of a point  $P$  is its reflection  $-P$ .

The coefficients  $a, b$  be part of

- $\mathbb{R}$  Real numbers
- $\mathbb{Q}$  Rational numbers
- $\mathbb{C}$  Complex numbers
- $\mathbb{Z}/p\mathbb{Z}$  Finite field

## 1.2 Addition

Given two points  $P, Q \in E$  we can describe a unique third point.

We take the line that intersects  $P$  and  $Q$ , the opposite of the third intersection with the curve is out point.

$$P + Q = -R$$

If  $P = Q$ , the intersection line will be given by the tangent at that point.

If  $P = -Q$ ,  $P + Q = O$ .

If  $P = -P$  (inflection point, the concavity of the curve changes)  $R = P$ ,  $P + P = -P = P$ .

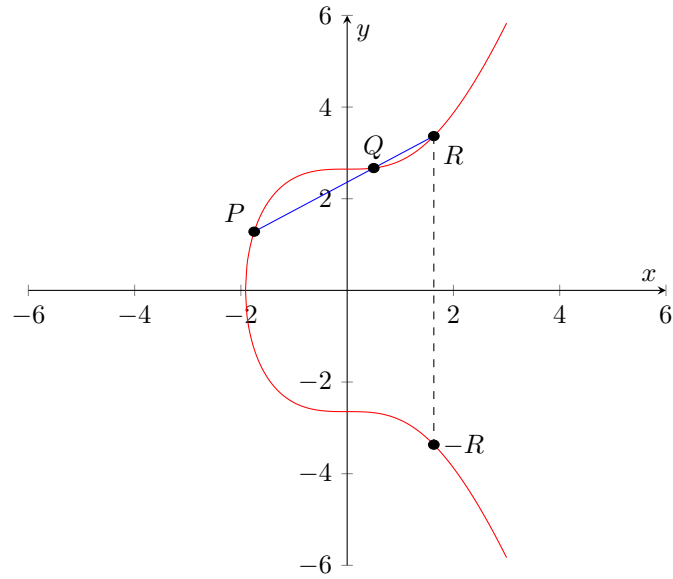
We consider  $-O$  to be  $O$ .

The intersection line  $mx + q$  is given by

$$m = \frac{P_y - Q_y}{P_x - Q_x}$$

and

$$q = P_y - mP_x$$



### 1.3 Scalar Multiplication

Given a point  $P \in E$ , multiplying  $kP$  where  $k \in \mathbb{Z}$  is equivalent to adding  $P$  to itself  $k$  times. Computing  $2P$  is the equivalent of  $P + P$  which can be calculated as  $P + Q = -R$ .

## 2 Discrete logarithm problem

### 2.1 Definition

Given an elliptic curve  $E$  and a point  $P \in E$ , we consider

$$kP = Q, \quad k \in \mathbb{Z}$$

Given the value of  $P$  and  $Q$  it is a hard problem to find the value of  $k$ .

We can use many ECC multiplication algorithms to compute  $kP$ , but reversing this operation for big values of  $k$  is not an easy task.

Furthermore, given  $k_1$  and  $k_2$ , we notice that

$$\begin{aligned} k_1(k_2P) &= k_2(k_1P) \\ &= (k_1 + k_2)P \end{aligned}$$

It does not matter if we first multiply  $P$  by  $k_1$  and then  $k_2$  or viceversa,  $P$  will always be multiplied  $k_1 + k_2$  times.

### 2.2 Diffie–Hellman

We can use the scalar multiplication function with the Diffie–Hellman method.

We define an elliptic curve  $E$  over a finite field  $\mathbb{F}_p$ .

The *client* and the *server* publicly establish the domain parameters

- $p$  The field that the curve is defined over  $\mathbb{F}_p \pmod{p}$ .
- $a$  The parameter  $a$  of the elliptic curve equation.
- $b$  The parameter  $b$  of the elliptic curve equation.
- $G$  The generator, a fixed point  $G \in E$ .
- $n$  The prime order of  $G$ , the smallest prime such that  $kG = O$ .  
In order words, the number of points that can be generated multiplying  $G$  with itself.
- $h$  The cofactor, the ratio between the amount of points in  $E$  and the prime order of  $G$ .  
Ideally we would want  $h = 1$ , which means that the points are well-distributed.

We then proceed with the Diffie–Hellman key exchange method.

1. The *client* generates a private key  $k_c \in \mathbb{Z}$  such that  $1 \leq k_c \leq n - 1$ .
2. The *server* generates a private key  $k_s \in \mathbb{Z}$  such that  $1 \leq k_s \leq n - 1$ .
3. The *client* computes a public key  $P_c = k_c G$ .
4. The *server* computes a public key  $P_s = k_s G$ .
5. The two parts publicly exchange the public keys.
6. The *client* computes  $R = k_c P_s$ .
7. The *server* computes  $R = k_s P_c$ .

Now both parts share the same secret point  $R \in E$ .