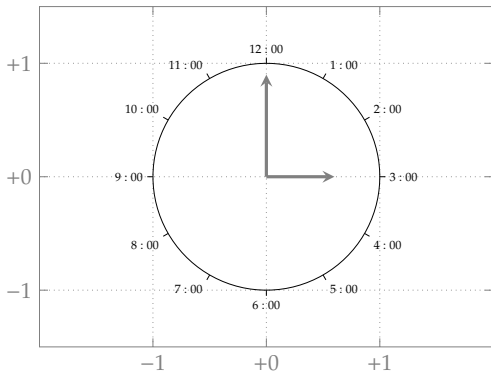


- ▶ **Agenda:** explore **elliptic curves**  $\leadsto$  **Elliptic Curve Cryptography (ECC)** via
  1. an “in theory”, i.e., Mathematics-oriented perspective, and
  2. an “in practice”, i.e., implementation-oriented perspective,with a focus somewhat more on “EC” than “C” throughout!
- ▶ **Caveat!**
  - ~ 2 hours  $\Rightarrow$  introductory, and (very) selective (versus definitive) coverage.

## Part 1: in theory (1)

An ECCHacks-based primer [1]: “from geometry to group theory”

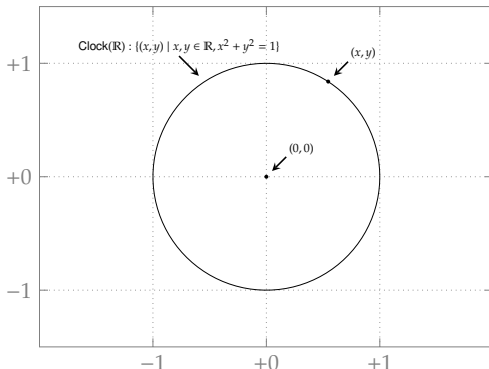
- **Idea:** consider a clock face.



## Part 1: in theory (1)

An ECCHacks-based primer [1]: “from geometry to group theory”

- **Idea:** consider a clock-face unit circle (i.e., a *non-elliptic* curve)



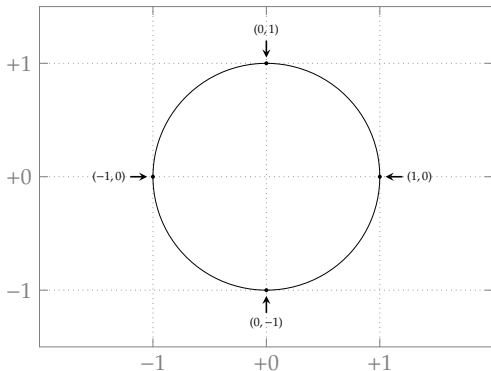
where we can describe points in Cartesian form, namely

$$P = (P_x, P_y).$$

## Part 1: in theory (1)

An ECCHacks-based primer [1]: “from geometry to group theory”

- **Idea:** consider a clock-face unit circle (i.e., a *non-elliptic* curve)



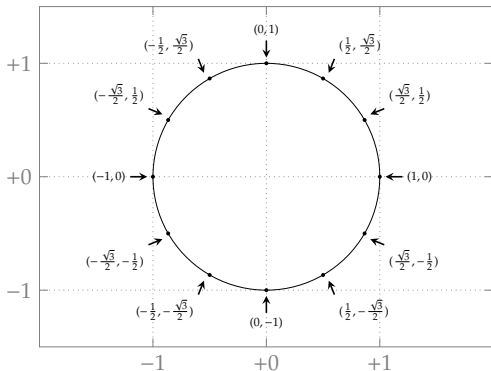
where we can describe points in Cartesian form, namely

$$P = (P_x, P_y).$$

## Part 1: in theory (1)

An ECCHacks-based primer [1]: “from geometry to group theory”

- **Idea:** consider a clock-face unit circle (i.e., a *non-elliptic curve*)



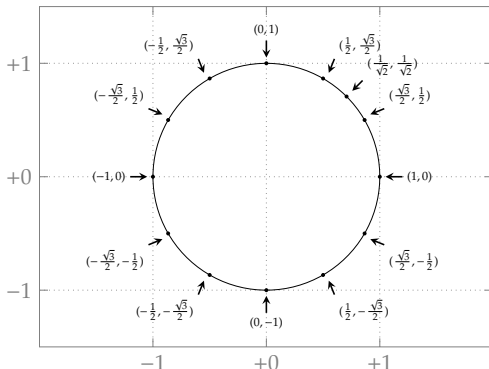
where we can describe points in Cartesian form, namely

$$P = (P_x, P_y).$$

## Part 1: in theory (1)

An ECCHacks-based primer [1]: “from geometry to group theory”

- **Idea:** consider a clock-face unit circle (i.e., a *non-elliptic curve*)



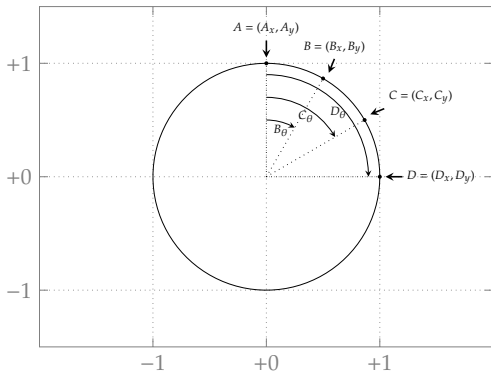
where we can describe points in Cartesian form, namely

$$P = (P_x, P_y).$$

## Part 1: in theory (1)

An ECCHacks-based primer [1]: “from geometry to group theory”

- **Idea:** consider a clock-face unit circle (i.e., a *non-elliptic* curve)



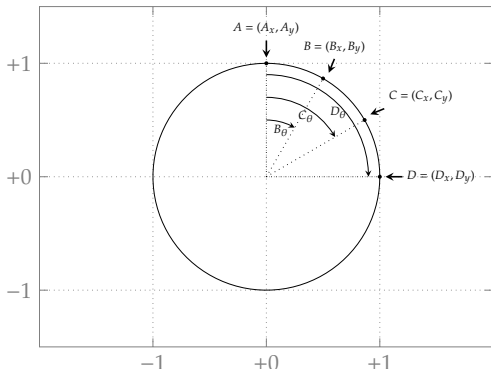
where we can describe points in Cartesian *or* parametric form, namely

$$P = (P_x, P_y) = (\sin P_\theta, \cos P_\theta).$$

## Part 1: in theory (1)

An ECCHacks-based primer [1]: “from geometry to group theory”

- **Idea:** consider a clock-face unit circle (i.e., a *non-elliptic* curve)



where we can define *addition* of points, such that

$$\begin{aligned} R = (R_x, R_y) &= P \oplus Q = (\sin(P_\theta + Q_\theta), \cos(P_\theta + Q_\theta)) \\ &= (\sin P_\theta \cos Q_\theta + \cos P_\theta \sin Q_\theta, \cos P_\theta \cos Q_\theta - \sin P_\theta \sin Q_\theta) \end{aligned}$$



## Part 1: in theory (2)

An ECCHacks-based primer [1]: “from geometry to group theory”

### Theorem

The set of “o’clock points”  $O'Clock(\mathbb{R}) \subset Clock(\mathbb{R})$  forms a group under  $\oplus$  with identity element “12:00”.

### ► “Proof”:

	Q												
	9 : 00	8 : 00	10 : 00	7 : 00	11 : 00	6 : 00	12 : 00	5 : 00	1 : 00	4 : 00	2 : 00	3 : 00	
9 : 00	6 : 00	5 : 00	7 : 00	4 : 00	8 : 00	3 : 00	9 : 00	2 : 00	10 : 00	1 : 00	11 : 00	12 : 00	
8 : 00	5 : 00	4 : 00	6 : 00	3 : 00	7 : 00	2 : 00	8 : 00	1 : 00	9 : 00	12 : 00	10 : 00	11 : 00	
10 : 00	7 : 00	6 : 00	8 : 00	5 : 00	9 : 00	4 : 00	10 : 00	3 : 00	11 : 00	2 : 00	12 : 00	1 : 00	
7 : 00	4 : 00	3 : 00	5 : 00	2 : 00	6 : 00	1 : 00	7 : 00	12 : 00	8 : 00	11 : 00	9 : 00	10 : 00	
11 : 00	8 : 00	7 : 00	9 : 00	6 : 00	10 : 00	5 : 00	11 : 00	4 : 00	12 : 00	3 : 00	1 : 00	2 : 00	
6 : 00	3 : 00	2 : 00	4 : 00	1 : 00	5 : 00	12 : 00	6 : 00	11 : 00	7 : 00	10 : 00	8 : 00	9 : 00	
12 : 00	9 : 00	8 : 00	10 : 00	7 : 00	11 : 00	6 : 00	12 : 00	5 : 00	1 : 00	4 : 00	2 : 00	3 : 00	
5 : 00	2 : 00	1 : 00	3 : 00	12 : 00	4 : 00	11 : 00	5 : 00	10 : 00	6 : 00	9 : 00	7 : 00	8 : 00	
1 : 00	10 : 00	9 : 00	11 : 00	8 : 00	12 : 00	7 : 00	1 : 00	6 : 00	2 : 00	5 : 00	3 : 00	4 : 00	
4 : 00	1 : 00	12 : 00	2 : 00	11 : 00	3 : 00	10 : 00	4 : 00	9 : 00	5 : 00	8 : 00	6 : 00	7 : 00	
2 : 00	11 : 00	10 : 00	12 : 00	9 : 00	1 : 00	8 : 00	2 : 00	7 : 00	3 : 00	6 : 00	4 : 00	5 : 00	
3 : 00	12 : 00	11 : 00	1 : 00	10 : 00	2 : 00	9 : 00	3 : 00	8 : 00	4 : 00	7 : 00	5 : 00	6 : 00	

## Part 1: in theory (2)

An ECC HACKS-based primer [1]: “from geometry to group theory”

### Theorem

The set of “o’clock points”  $O\text{Clock}(\mathbb{R}) \subset \text{Clock}(\mathbb{R})$  forms a group under  $\oplus$  with identity element “12:00”.

#### ► “Proof”:

	Q											
	9 : 00	8 : 00	10 : 00	7 : 00	11 : 00	6 : 00	12 : 00	5 : 00	1 : 00	4 : 00	2 : 00	3 : 00
9 : 00	6 : 00	5 : 00	7 : 00	4 : 00	8 : 00	3 : 00	9 : 00	2 : 00	10 : 00	1 : 00	11 : 00	12 : 00
8 : 00	5 : 00	4 : 00	6 : 00	3 : 00	7 : 00	2 : 00	8 : 00	1 : 00	9 : 00	12 : 00	10 : 00	11 : 00
10 : 00	7 : 00	6 : 00	8 : 00	5 : 00	9 : 00	4 : 00	10 : 00	3 : 00	11 : 00	2 : 00	12 : 00	1 : 00
7 : 00	4 : 00	3 : 00	5 : 00	2 : 00	6 : 00	1 : 00	7 : 00	12 : 00	8 : 00	11 : 00	9 : 00	10 : 00
11 : 00	8 : 00	7 : 00	9 : 00	6 : 00	10 : 00	5 : 00	11 : 00	4 : 00	12 : 00	3 : 00	1 : 00	2 : 00
6 : 00	3 : 00	2 : 00	4 : 00	1 : 00	5 : 00	12 : 00	6 : 00	11 : 00	7 : 00	10 : 00	8 : 00	9 : 00
12 : 00	9 : 00	8 : 00	10 : 00	7 : 00	11 : 00	6 : 00	12 : 00	5 : 00	1 : 00	4 : 00	2 : 00	3 : 00
5 : 00	2 : 00	1 : 00	3 : 00	12 : 00	4 : 00	11 : 00	5 : 00	10 : 00	6 : 00	9 : 00	7 : 00	8 : 00
1 : 00	10 : 00	9 : 00	11 : 00	8 : 00	12 : 00	7 : 00	1 : 00	6 : 00	2 : 00	5 : 00	3 : 00	4 : 00
4 : 00	1 : 00	12 : 00	2 : 00	11 : 00	3 : 00	10 : 00	4 : 00	9 : 00	5 : 00	8 : 00	6 : 00	7 : 00
2 : 00	11 : 00	10 : 00	12 : 00	9 : 00	1 : 00	8 : 00	2 : 00	7 : 00	3 : 00	6 : 00	4 : 00	5 : 00
3 : 00	12 : 00	11 : 00	1 : 00	10 : 00	2 : 00	9 : 00	3 : 00	8 : 00	4 : 00	7 : 00	5 : 00	6 : 00

#### ► Example:

$$\begin{aligned} \text{“1:00”} \oplus \text{“2:00”} &= (\sin 30, \cos 30) \oplus (\sin 60, \cos 60) \\ &= (\sin(30 + 60), \cos(30 + 60)) \\ &= (\sin 90, \cos 90) \\ &= \text{“3:00”} \end{aligned}$$

## Part 1: in theory (2)

An ECC HACKS-based primer [1]: “from geometry to group theory”

### Theorem

The set of “o’clock points”  $O'Clock(\mathbb{R}) \subset Clock(\mathbb{R})$  forms a group under  $\oplus$  with identity element “12:00”.

#### ► “Proof”:

	Q												
	9 : 00	8 : 00	10 : 00	7 : 00	11 : 00	6 : 00	12 : 00	5 : 00	1 : 00	4 : 00	2 : 00	3 : 00	
9 : 00	6 : 00	5 : 00	7 : 00	4 : 00	8 : 00	3 : 00	9 : 00	2 : 00	10 : 00	1 : 00	11 : 00	12 : 00	
8 : 00	5 : 00	4 : 00	6 : 00	3 : 00	7 : 00	2 : 00	8 : 00	1 : 00	9 : 00	12 : 00	10 : 00	11 : 00	
10 : 00	7 : 00	6 : 00	8 : 00	5 : 00	9 : 00	4 : 00	10 : 00	3 : 00	11 : 00	2 : 00	12 : 00	1 : 00	
7 : 00	4 : 00	3 : 00	5 : 00	2 : 00	6 : 00	1 : 00	7 : 00	12 : 00	8 : 00	11 : 00	9 : 00	10 : 00	
11 : 00	8 : 00	7 : 00	9 : 00	6 : 00	10 : 00	5 : 00	11 : 00	4 : 00	12 : 00	3 : 00	1 : 00	2 : 00	
6 : 00	3 : 00	2 : 00	4 : 00	1 : 00	5 : 00	12 : 00	6 : 00	11 : 00	7 : 00	10 : 00	8 : 00	9 : 00	
12 : 00	9 : 00	8 : 00	10 : 00	7 : 00	11 : 00	6 : 00	12 : 00	5 : 00	1 : 00	4 : 00	2 : 00	3 : 00	
5 : 00	2 : 00	1 : 00	3 : 00	12 : 00	4 : 00	11 : 00	5 : 00	10 : 00	6 : 00	9 : 00	7 : 00	8 : 00	
1 : 00	10 : 00	9 : 00	11 : 00	8 : 00	12 : 00	7 : 00	1 : 00	6 : 00	2 : 00	5 : 00	3 : 00	4 : 00	
4 : 00	1 : 00	12 : 00	2 : 00	11 : 00	3 : 00	10 : 00	4 : 00	9 : 00	5 : 00	8 : 00	6 : 00	7 : 00	
2 : 00	11 : 00	10 : 00	12 : 00	9 : 00	1 : 00	8 : 00	2 : 00	7 : 00	3 : 00	6 : 00	4 : 00	5 : 00	
3 : 00	12 : 00	11 : 00	1 : 00	10 : 00	2 : 00	9 : 00	3 : 00	8 : 00	4 : 00	7 : 00	5 : 00	6 : 00	

#### ► Example:

$$\begin{aligned} \text{“1:00”} \oplus \text{“12:00”} &= (\sin 30, \cos 30) \oplus (\sin 0, \cos 0) \\ &= (\sin(30 + 0), \cos(30 + 0)) \\ &= (\sin 30, \cos 30) \\ &= \text{“1:00”} \end{aligned}$$

## Part 1: in theory (3)

An ECCHacks-based primer [1]: “from geometry to group theory”

► **Idea:** eliminate non-discrete aspect of  $\text{Clock}(\mathbb{R})$ , by

1. considering

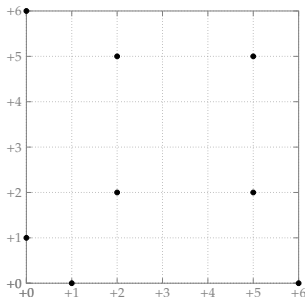
$$\text{Clock}(\mathbb{F}_q) : \{(x, y) \mid x, y \in \mathbb{F}_q, x^2 + y^2 = 1\},$$

then

2. translating

$$\begin{aligned} R = (R_x, R_y) &= P \oplus Q = (\sin P_\theta \cos Q_\theta + \cos P_\theta \sin Q_\theta, \cos P_\theta \cos Q_\theta - \sin P_\theta \sin Q_\theta) \\ &\equiv (P_x Q_y + P_y Q_x, P_y Q_y - P_x Q_x) \end{aligned}$$

such that if  $q = 7$ , for example, we *naturally* get a set of discrete points:



## Part 1: in theory (3)

An ECCHacks-based primer [1]: “from geometry to group theory”

► **Idea:** eliminate non-discrete aspect of  $\text{Clock}(\mathbb{R})$ , by

1. considering

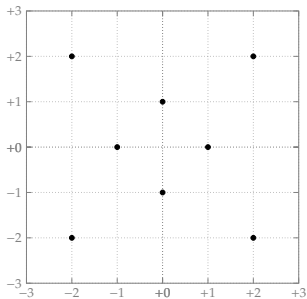
$$\text{Clock}(\mathbb{F}_q) : \{(x, y) \mid x, y \in \mathbb{F}_q, x^2 + y^2 = 1\},$$

then

2. translating

$$\begin{aligned} R = (R_x, R_y) &= P \oplus Q = (\sin P_\theta \cos Q_\theta + \cos P_\theta \sin Q_\theta, \cos P_\theta \cos Q_\theta - \sin P_\theta \sin Q_\theta) \\ &\equiv (P_x Q_y + P_y Q_x, P_y Q_y - P_x Q_x) \end{aligned}$$

such that if  $q = 7$ , for example, we *naturally* get a set of discrete points:



## Part 1: in theory (4)

An ECCHacks-based primer [1]: “from geometry to group theory”

### Theorem

The set  $\text{Clock}(\mathbb{F}_7)$  forms a group under  $\oplus$  with identity element  $(0, 1)$ .

► “Proof”:

	Q								
	(0, 1)	(0, 6)	(1, 0)	(2, 2)	(2, 5)	(5, 2)	(5, 5)	(6, 0)	
P	(0, 1)	(0, 1)	(0, 6)	(1, 0)	(2, 2)	(2, 5)	(5, 2)	(5, 5)	(6, 0)
	(0, 6)	(0, 6)	(0, 1)	(6, 0)	(5, 5)	(5, 2)	(2, 5)	(2, 2)	(1, 0)
	(1, 0)	(1, 0)	(6, 0)	(0, 6)	(2, 5)	(5, 5)	(2, 2)	(5, 2)	(0, 1)
	(2, 2)	(2, 2)	(5, 5)	(2, 5)	(1, 0)	(0, 6)	(0, 1)	(6, 0)	(5, 2)
	(2, 5)	(2, 5)	(5, 2)	(5, 5)	(0, 6)	(6, 0)	(1, 0)	(0, 1)	(2, 2)
	(5, 2)	(5, 2)	(2, 5)	(2, 2)	(0, 1)	(1, 0)	(6, 0)	(0, 6)	(5, 5)
	(5, 5)	(5, 5)	(2, 2)	(5, 2)	(6, 0)	(0, 1)	(0, 6)	(1, 0)	(2, 5)
	(6, 0)	(6, 0)	(1, 0)	(0, 1)	(5, 2)	(2, 2)	(5, 5)	(2, 5)	(0, 6)

## Part 1: in theory (4)

An ECCHacks-based primer [1]: “from geometry to group theory”

### Theorem

The set  $\text{Clock}(\mathbb{F}_7)$  forms a group under  $\oplus$  with identity element  $(0, 1)$ .

► “Proof”:

	Q								
	(0,1)	(0,6)	(1,0)	(2,2)	(2,5)	(5,2)	(5,5)	(6,0)	
P	(0,1)	(0,1)	(0,6)	(1,0)	(2,2)	(2,5)	(5,2)	(5,5)	(6,0)
	(0,6)	(0,6)	(0,1)	(6,0)	(5,5)	(5,2)	(2,5)	(2,2)	(1,0)
	(1,0)	(1,0)	(6,0)	(0,6)	(2,5)	(5,5)	(2,2)	(5,2)	(0,1)
	(2,2)	(2,2)	(5,5)	(2,5)	(1,0)	(0,6)	(0,1)	(6,0)	(5,2)
	(2,5)	(2,5)	(5,2)	(5,5)	(0,6)	(6,0)	(1,0)	(0,1)	(2,2)
	(5,2)	(5,2)	(2,5)	(2,2)	(0,1)	(1,0)	(6,0)	(0,6)	(5,5)
	(5,5)	(5,5)	(2,2)	(5,2)	(6,0)	(0,1)	(0,6)	(1,0)	(2,5)
	(6,0)	(6,0)	(1,0)	(0,1)	(5,2)	(2,2)	(5,5)	(2,5)	(0,6)

► Example:

$$\begin{aligned}(0,6) \oplus (1,0) &= (0 \times 0 + 6 \times 1, 6 \times 0 + 0 \times 1) \\ &= (6,0)\end{aligned}$$

## Part 1: in theory (4)

An ECCHacks-based primer [1]: “from geometry to group theory”

### Theorem

The set  $\text{Clock}(\mathbb{F}_7)$  forms a group under  $\oplus$  with identity element  $(0, 1)$ .

► “Proof”:

	Q								
	(0,1)	(0,6)	(1,0)	(2,2)	(2,5)	(5,2)	(5,5)	(6,0)	
P	(0,1)	(0,1)	(0,6)	(1,0)	(2,2)	(2,5)	(5,2)	(5,5)	(6,0)
	(0,6)	(0,6)	(0,1)	(6,0)	(5,5)	(5,2)	(2,5)	(2,2)	(1,0)
	(1,0)	(1,0)	(6,0)	(0,6)	(2,5)	(5,5)	(2,2)	(5,2)	(0,1)
	(2,2)	(2,2)	(5,5)	(2,5)	(1,0)	(0,6)	(0,1)	(6,0)	(5,2)
	(2,5)	(2,5)	(5,2)	(5,5)	(0,6)	(6,0)	(1,0)	(0,1)	(2,2)
	(5,2)	(5,2)	(2,5)	(2,2)	(0,1)	(1,0)	(6,0)	(0,6)	(5,5)
	(5,5)	(5,5)	(2,2)	(5,2)	(6,0)	(0,1)	(0,6)	(1,0)	(2,5)
	(6,0)	(6,0)	(1,0)	(0,1)	(5,2)	(2,2)	(5,5)	(2,5)	(0,6)

► Example:

$$\begin{aligned}(0,6) \oplus (0,1) &= (0 \times 1 + 6 \times 0, 6 \times 1 + 0 \times 0) \\ &= (0,6)\end{aligned}$$



#### Definition

1. An **elliptic curve**  $E$  over the field  $K$  is defined by the general (or “long”) **Weierstraß equation**, namely

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

for  $a_i \in K$ .

2. The  **$K$ -rational** set of points on such an  $E$  is

$$E(K) : \{(x, y) \mid x, y \in K, y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6\} \cup \{O\}$$

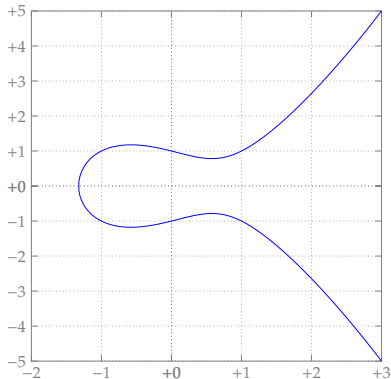
i.e., the set of points which satisfy the curve equation plus an extra **point at infinity**.

## Part 1: in theory (8)

General elliptic curves over  $K$

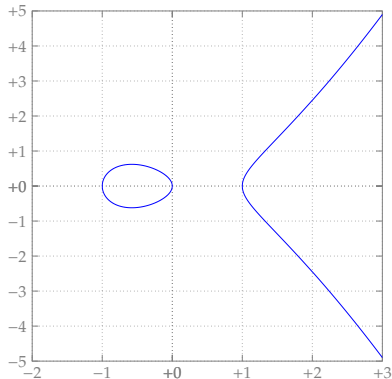
### Example

Consider  $E(\mathbb{R})$  for  $a_1 = 0$ ,  $a_2 = 0$ ,  $a_3 = 0$ ,  $a_4 = -1$ , and  $a_6 = 1$ , meaning  $\Delta = -368$  and  $j(E) = -6912/23$ :



### Example

Consider  $E(\mathbb{R})$  for  $a_1 = 0$ ,  $a_2 = 0$ ,  $a_3 = 0$ ,  $a_4 = -1$ , and  $a_6 = 0$ , meaning  $\Delta = 64$  and  $j(E) = 1728$ :



## Definition

Provided  $K$  is algebraically closed, a line drawn through two  $K$ -rational points  $P$  and  $Q$  will always intersect  $E$  at a third  $K$ -rational point  $T$ . The **elliptic curve group law** states that

$$P \oplus Q \oplus T = \mathcal{O}$$

i.e., the sum of the three points of intersection is  $\mathcal{O}$ .

► **Idea:** the **chord-tangent** (or **line-tangent**) process affords a group operation:

► Consider  $P, Q \in E(K)$ :

- if  $P \neq Q$  then a line drawn between them will intersect  $E$  at  $T$ ,
- if  $P = Q$  then a line drawn between them is a tangent to  $E$ .

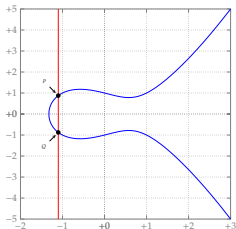
► The group operation is then just geometry:

1. draw a line between  $P$  and  $Q$ , let  $T$  be the third point of intersection on  $E$ , then
2. draw a line between  $T$  and  $\mathcal{O}$ , let  $R = P \oplus Q$  be the third point of intersection on  $E$ .

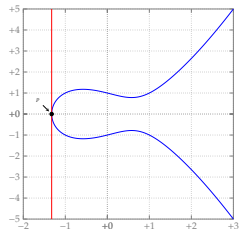
# Part 1: in theory (10)

General elliptic curves over  $K$

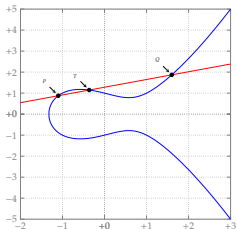
Case #1:  $P \oplus Q \oplus \mathcal{O} = \mathcal{O} \Rightarrow P = -Q$



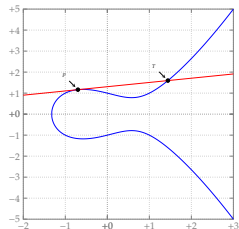
Case #2:  $P \oplus P \oplus \mathcal{O} = \mathcal{O}$



Case #3:  $P \oplus Q \oplus T = \mathcal{O} \Rightarrow T = -(P \oplus Q)$



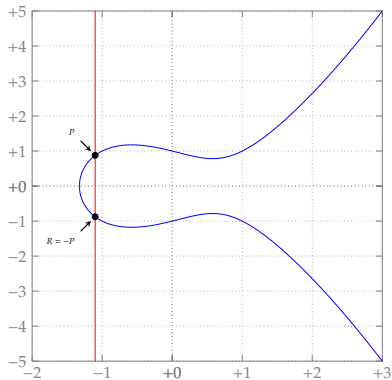
Case #4:  $P \oplus P \oplus T = \mathcal{O} \Rightarrow T = -(P \oplus P)$



## Part 1: in theory (11)

General elliptic curves over  $K$

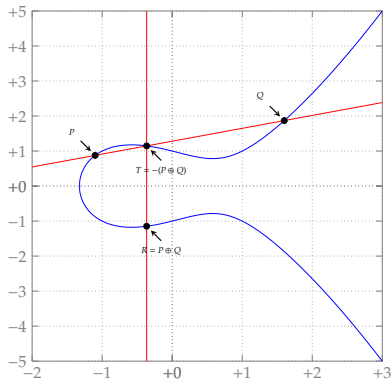
### Algorithm (affine point negation)



► For  $P = (P_x, P_y)$ ,  $R = (R_x, R_y) = -P$  is computed via

$$\begin{aligned}R_x &= P_x \\R_y &= -P_y - a_1 P_x - a_3\end{aligned}$$

## Algorithm (affine point addition)



- ▶ For  $P = (P_x, P_y)$  and  $Q = (Q_x, Q_y)$ , let

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

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

- ▶  $R = (R_x, R_y) = P \oplus Q$  is computed via

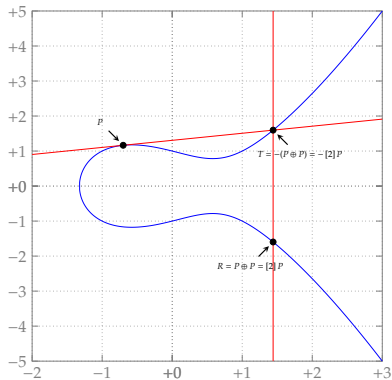
$$R_x = \lambda^2 + a_1 \lambda - a_2 - P_x - Q_x$$

$$R_y = -(\lambda + a_1)R_x - \mu - a_3$$

## Part 1: in theory (13)

### General elliptic curves over $K$

#### Algorithm (affine point doubling)



- ▶ For  $P = (P_x, P_y)$ , let

$$\lambda = \frac{3P_x^2 + 2a_2P_x + a_4 - a_1P_y}{2P_y + a_1P_x + a_3}$$

$$\mu = \frac{-P_x^3 + a_4P_x + 2a_6 - a_3P_y}{2P_y + a_1P_x + a_3}$$

- ▶  $R = (R_x, R_y) = P \oplus P = [2]P$ , is computed via

$$R_x = \lambda^2 + a_1\lambda - a_2 - P_x - Q_x$$

$$R_y = -(\lambda + a_1)R_x - \mu - a_3$$

## Part 1: in theory (14)

Cryptographic elliptic curves over  $\mathbb{F}_q$

- **Concept:** cryptographic use-cases set

$$K = \mathbb{F}_q$$

i.e., use elliptic curves over a **finite field** such as

1.  $q = p \Rightarrow$  large prime characteristic field  $\mathbb{F}_p$
2.  $q = 2^m \Rightarrow$  characteristic two (or binary extension) field  $\mathbb{F}_{2^m}$

meaning we can

1. specialise the long Weierstraß equation and hence the related group operation,
2. represent, and compute with, group elements efficiently, and
3. reason more directly about security of such curves.

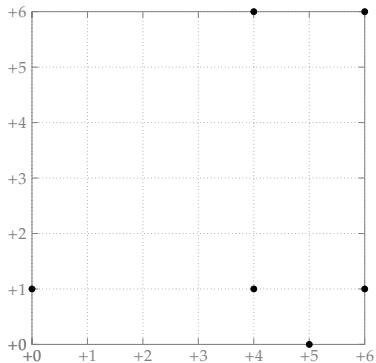


# Part 1: in theory (15)

Cryptographic elliptic curves over  $\mathbb{F}_q$

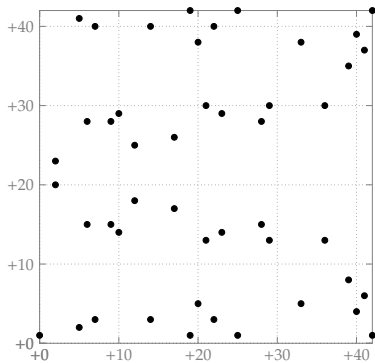
## Example

Consider  $E(\mathbb{F}_7)$  for  $a_1 = 0, a_2 = 0, a_3 = 0, a_4 = 1$ , and  $a_6 = 3$ , meaning  $\Delta = 3$  and  $j(E) = 5$ :



## Example

Consider  $E(\mathbb{F}_{43})$  for  $a_1 = 0, a_2 = 0, a_3 = 0, a_4 = 1$ , and  $a_6 = 3$ , meaning  $\Delta = 4$  and  $j(E) = 1$ :

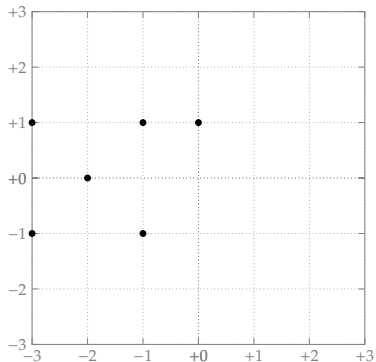


# Part 1: in theory (15)

Cryptographic elliptic curves over  $\mathbb{F}_q$

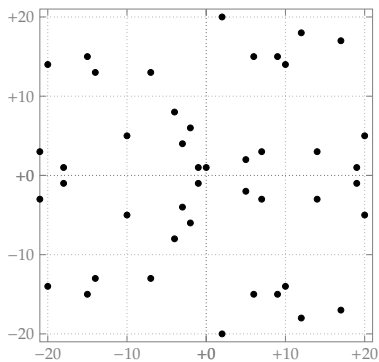
## Example

Consider  $E(\mathbb{F}_7)$  for  $a_1 = 0, a_2 = 0, a_3 = 0, a_4 = 1$ , and  $a_6 = 3$ , meaning  $\Delta = 3$  and  $j(E) = 5$ :



## Example

Consider  $E(\mathbb{F}_{43})$  for  $a_1 = 0, a_2 = 0, a_3 = 0, a_4 = 1$ , and  $a_6 = 3$ , meaning  $\Delta = 4$  and  $j(E) = 1$ :



## Part 1: in theory (16)

Cryptographic elliptic curves over  $\mathbb{F}_q$

### Algorithm (specialisation for $q = p$ )

1. Specialise the Weierstraß equation to

$$E : y^2 = x^3 + a_4x + a_6.$$

2. Given  $P = (P_x, P_y)$ , to compute

$$R = (R_x, R_y) = -P$$

we set

$$\begin{aligned} R_x &= P_x \\ R_y &= -P_y \end{aligned}$$

3. Given  $P = (P_x, P_y)$  and  $Q = (Q_x, Q_y)$ , to compute

$$R = (R_x, R_y) = P \oplus Q$$

we set

$$\lambda = \begin{cases} \frac{Q_y - P_y}{Q_x - P_x} & \text{if } P \neq Q \\ \frac{3P_x^2 + a_4}{2P_y} & \text{if } P = Q \end{cases}$$

then

$$\begin{aligned} R_x &= \lambda^2 - P_x - Q_x \\ R_y &= \lambda(P_x - R_x) - P_y \end{aligned}$$

# Part 1: in theory (16)

Cryptographic elliptic curves over  $\mathbb{F}_q$

## Algorithm (specialisation for $q = 2^m$ )

1. Specialise the Weierstraß equation to

$$E : y^2 + xy = x^3 + a_2x^2 + a_6.$$

2. Given  $P = (P_x, P_y)$ , to compute

$$R = (R_x, R_y) = -P$$

we set

$$\begin{aligned} R_x &= P_x \\ R_y &= P_y + P_x \end{aligned}$$

3. Given  $P = (P_x, P_y)$  and  $Q = (Q_x, Q_y)$ , to compute

$$R = (R_x, R_y) = P \oplus Q$$

we set

$$\lambda = \begin{cases} \frac{Q_y + P_y}{Q_x + P_x} & \text{if } P \neq Q \\ \frac{P_y + P_x}{P_x} & \text{if } P = Q \end{cases}$$

then

$$\begin{aligned} R_x &= \lambda^2 + \lambda + a_2 + P_x + Q_x \\ R_y &= \lambda(P_x + R_x) + R_x + P_y \end{aligned}$$

## Example

Let  $K = \mathbb{F}_7$ , and consider  $E(K)$  for  $a_1 = 0, a_2 = 0, a_3 = 0, a_4 = 1$ , and  $a_6 = 3$ , such that

$$E(K) = \{O, (4, 1), (4, 6), (5, 0), (6, 1), (6, 6)\}$$

and hence  $|E(K)| = 6$ . If  $P = (P_x, P_y) = (4, 1)$  and  $Q = (Q_x, Q_y) = (5, 0)$  then  $R = (R_x, R_y) = P \oplus Q$  is given by

$$\begin{aligned} \lambda &= \frac{Q_y - P_y}{Q_x - P_x} = \frac{0 - 1}{5 - 4} = 6 \pmod{7} \\ R_x &= \lambda^2 - Q_x - P_x = 6^2 - 4 - 5 = 6 \pmod{7} \\ R_y &= \lambda(P_x - R_x) - P_y = 6 \times (4 - 6) - 1 = 1 \pmod{7} \end{aligned}$$

and in fact we can describe the entire group operation as

		Q					
		O	(4, 1)	(4, 6)	(5, 0)	(6, 1)	(6, 6)
P	O	O	(4, 1)	(4, 6)	(5, 0)	(6, 1)	(6, 6)
	(4, 1)	(4, 1)	(6, 6)	O	(6, 1)	(4, 6)	(5, 0)
	(4, 6)	(4, 6)	O	(6, 1)	(6, 6)	(5, 0)	(4, 1)
	(5, 0)	(5, 0)	(6, 1)	(6, 6)	O	(4, 1)	(4, 6)
	(6, 1)	(6, 1)	(4, 6)	(5, 0)	(4, 1)	(6, 6)	O
	(6, 6)	(6, 6)	(5, 0)	(4, 1)	(4, 6)	O	(6, 1)

# Part 1: in theory (18)

Cryptographic elliptic curves over  $\mathbb{F}_q$

## Example

Let  $K = \mathbb{F}_2[x]/x^2 + x + 1$ , and consider  $E(K)$  for  $a_1 = 1, a_2 = 0, a_3 = 0, a_4 = 0$ , and  $a_6 = 1$ , such that

$$E(K) = \{O, (0, 1), (1, 0), (1, 1), (x, 0), (x, x), (x + 1, 0), (x + 1, x + 1)\}$$

and hence  $|E(K)| = 8$ . If  $P = (P_x, P_y) = (x + 1, 0)$  and  $Q = (Q_x, Q_y) = (x, x)$  then  $R = (R_x, R_y) = P \oplus Q$  is given by

$$\begin{aligned} \lambda &= \frac{Q_y - P_y}{Q_x - P_x} = \frac{x - 0}{x - x + 1} = x \pmod{x^2 + x + 1} \\ R_x &= \lambda^2 + \lambda + a_2 + P_x + Q_x = x^2 + x + 0 + (x + 1) + x = 0 \pmod{x^2 + x + 1} \\ R_y &= \lambda(P_x + R_x) + R_y + P_y = x \times ((x + 1) + 0) + 0 + 0 = 1 \pmod{x^2 + x + 1} \end{aligned}$$

and in fact we can describe the entire group operation as

		Q							
		O	(0,1)	(1,0)	(1,1)	(x,0)	(x,x)	(x+1,0)	(x+1,x+1)
P	O	O	(0,1)	(1,0)	(1,1)	(x,0)	(x,x)	(x+1,0)	(x+1,x+1)
	(0,1)	(0,1)	O	(1,1)	(1,0)	(x+1,0)	(x+1,x+1)	(x,0)	(x,x)
	(1,0)	(1,0)	(1,1)	(0,1)	O	(x+1,x+1)	(x,0)	(x,x)	(x+1,0)
	(1,1)	(1,1)	(1,0)	O	(0,1)	(x,x)	(x+1,0)	(x+1,x+1)	(x,0)
	(x,0)	(x,0)	(x+1,0)	(x+1,x+1)	(x,x)	(1,0)	O	(1,1)	(0,1)
	(x,x)	(x,x)	(x+1,x+1)	(x,0)	(x+1,0)	O	(1,1)	(0,1)	(1,0)
	(x+1,0)	(x+1,0)	(x,0)	(x,x)	(x+1,x+1)	(1,1)	(0,1)	(x,0)	O
	(x+1,x+1)	(x+1,x+1)	(x,x)	(x+1,0)	(x,0)	(0,1)	(1,0)	O	(1,1)

## Part 1: in theory (19)

Cryptographic elliptic curves over  $\mathbb{F}_q$

### Definition

Consider an elliptic curve  $E(\mathbb{F}_q)$  and points  $P \in E(\mathbb{F}_q)$  and  $Q \in \langle P \rangle$  of order  $n$ . Given

$$Q = [l]P,$$

$l \in \{0, 1, \dots, n-1\}$  is called the **Elliptic Curve Discrete Logarithm (EC-DL)** of  $Q$  to the base  $P$ ; the corresponding **Elliptic Curve Discrete Logarithm Problem (EC-DLP)** is to compute  $l$  given  $P$  and  $Q$ , i.e., to compute  $l = \log_P Q$ .

## Part 1: in theory (19)

Cryptographic elliptic curves over  $\mathbb{F}_q$

### Definition

Consider an elliptic curve  $E(\mathbb{F}_q)$  and points  $P \in E(\mathbb{F}_q)$  and  $Q \in \langle P \rangle$  of order  $n$ . Given

$$Q = [l]P,$$

$l \in \{0, 1, \dots, n-1\}$  is called the **Elliptic Curve Discrete Logarithm (EC-DL)** of  $Q$  to the base  $P$ ; the corresponding **Elliptic Curve Discrete Logarithm Problem (EC-DLP)** is to compute  $l$  given  $P$  and  $Q$ , i.e., to compute  $l = \log_P Q$ .

### ► We're done!

1. we can relax notation to  $+ \equiv \oplus$ , and say we have an *additive* group  $G^+ = (E(K), +)$ ,
2. we can define **scalar multiplication** as

$$Q = [l]P$$

i.e.,

$$Q = \underbrace{P + P + \dots + P}_{\text{total of } l \text{ terms}},$$

and

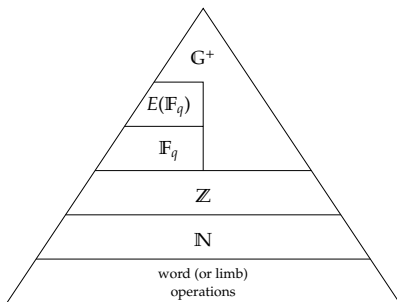
3. doing so allows us to pose a (EC-)DLP such that
  - given  $P$  and  $l$ , it is *easy* to compute  $Q$ , *but*
  - given  $P$  and  $Q$ , it is *hard* to compute  $l$  (on a suitable curve)and thus use it in a group-based scheme.



## Part 2: in practice (1)

### ► Challenge:

- given a functionality “stack”, i.e.,



bridge gap between what we have (bottom) and want (top),

- an **implementation strategy** for doing so must consider many

- goals : parameter set, functionality, ...
- metrics : latency, throughput, memory footprint, ...
- constraints : hardware versus software, data-path width, ...
-

## Part 2: in practice (2)

- **Motivation:** (simplified) comparison of DSA [5, Section 4] and EC-DSA [5, Section 6]

```
1 DSA.PARAMGEN begin
2   generate a prime  $q$ 
3   generate a prime  $p$  such that  $q \mid p - 1$ 
4   generate a  $G^X = \mathbb{F}_q = \langle g \rangle$  of order  $q$ 
5 end
```

```
1 EC-DSA.PARAMGEN begin
2
3
4   generate a  $G^+ = E(\mathbb{F}_q) = \langle G \rangle$  of order  $n$ 
5 end
```

raises the question of where/why the latter offers an advantage.

## Part 2: in practice (2)

- **Motivation:** (simplified) comparison of DSA [5, Section 4] and EC-DSA [5, Section 6]

```
1 DSA.KEYGEN begin
2   | select  $x \xleftarrow{\$} \{1, 2, \dots, q-1\}$ 
3   | compute  $y = g^x \pmod{p}$ 
4   | return  $(x, y)$ 
5 end
```

```
1 EC-DSA.KEYGEN begin
2   | select  $l \xleftarrow{\$} \{1, 2, \dots, n-1\}$ 
3   | compute  $Q = [l]G$ 
4   | return  $(l, Q)$ 
5 end
```

raises the question of where/why the latter offers an advantage.

## Part 2: in practice (2)

- **Motivation:** (simplified) comparison of DSA [5, Section 4] and EC-DSA [5, Section 6]

```
1 DSA.SIGN begin
2   select  $k \xleftarrow{\$} \{1, 2, \dots, q - 1\}$ 
3   compute  $h = H(m)$ 
4   compute  $r = (g^k \bmod p) \bmod q$ 
5   compute  $s = (k^{-1} \cdot (h + r \cdot x)) \bmod q$ 
6   return  $(r, s)$ 
7 end
```

```
1 EC-DSA.SIGN begin
2   select  $k \xleftarrow{\$} \{1, 2, \dots, n - 1\}$ 
3   compute  $h = \text{LSB}_{|n|}(H(m))$ 
4   compute  $r = R_x \bmod n$  where  $(R_x, R_y) = R = [k]G$ 
5   compute  $s = (k^{-1} \cdot (h + r \cdot l)) \bmod n$ 
6   return  $(r, s)$ 
7 end
```

raises the question of where/why the latter offers an advantage.

► **Motivation:** (simplified) comparison of DSA [5, Section 4] and EC-DSA [5, Section 6]

```
1 DSA.VERIFY begin
2   return false if  $0 \nmid r, s$  or  $r, s \nmid n$ 
3   compute  $h = H(m)$ 
4   compute  $u_1 = h \cdot s^{-1} \bmod q$ 
5   compute  $u_2 = r \cdot s^{-1} \bmod q$ 
6   compute  $v = (g^{u_1} \cdot y^{u_2} \bmod p) \bmod q$ 
7   return false if  $r \neq v$ , else return true
8 end
```

```
1 EC-DSA.VERIFY begin
2   return false if  $0 \nmid r, s$  or  $r, s \nmid n$ 
3   compute  $h = \text{LSB}_{|n|}(H(m))$ 
4   compute  $u_1 = h \cdot s^{-1} \bmod n$ 
5   compute  $u_2 = r \cdot s^{-1} \bmod n$ 
6   compute  $v = V_x \bmod n$  where  $(V_x, V_y) = V = [u_1]G + [u_2]Q$ 
7   return false if  $r \neq v$ , else return true
8 end
```

raises the question of where/why the latter offers an advantage.

## Part 2: in practice (3)

### Field parameters

- ▶ Idea: per ENISA [6, Table 3.6], e.g.,

Primitive	Parameter	Recommendation		
		Legacy	Near-term	Long-term
AES	$\log_2 k$	80	128	256
RSA	$\log_2 N$	1024	3072	15360
DLP	$\log_2 q$ (sub-group)	160	256	512
	$\log_2 p$ (group)	1024	3072	15360
EC-DLP	$\log_2 p$	160	256	512

EC-based groups can use a  $p$  which is

1. short, and
2. special-form.

## Algorithm (NIST-P-256-REDUCE, per Solinas [4, Example 3, Page 20])

**Input:** For  $w = 32$ -bit words, a 16-word integer product  $z = x \cdot y$  and the modulus  $p = 2^{256} - 2^{224} + 2^{192} + 2^{96} - 1$

**Output:** The result  $r = z \pmod{p}$

1. Form the nine, 8-word intermediate variables

$$\begin{aligned}
 S_0 &= \langle z_0, z_1, z_2, z_3, z_4, z_5, z_6, z_7 \rangle \\
 S_1 &= \langle 0, 0, 0, z_{11}, z_{12}, z_{13}, z_{14}, z_{15} \rangle \\
 S_2 &= \langle 0, 0, 0, z_{12}, z_{13}, z_{14}, z_{15}, 0 \rangle \\
 S_3 &= \langle z_8, z_9, z_{10}, 0, 0, 0, z_{14}, z_{15} \rangle \\
 S_4 &= \langle z_9, z_{10}, z_{11}, z_{13}, z_{14}, z_{15}, z_{13}, z_8 \rangle \\
 S_5 &= \langle z_{11}, z_{12}, z_{13}, 0, 0, 0, z_8, z_{10} \rangle \\
 S_6 &= \langle z_{12}, z_{13}, z_{14}, z_{15}, 0, 0, z_9, z_{11} \rangle \\
 S_7 &= \langle z_{13}, z_{14}, z_{15}, z_8, z_9, z_{10}, 0, z_{12} \rangle \\
 S_8 &= \langle z_{14}, z_{15}, 0, z_9, z_{10}, z_{11}, 0, z_{13} \rangle
 \end{aligned}$$

2. Compute

$$r = S_0 + 2S_1 + 2S_2 + S_3 + S_4 - S_5 - S_6 - S_7 - S_8 \pmod{p}.$$

3. Return  $0 \leq r < p$ .

## Example

Given  $p = 2^{256} - 2^{224} + 2^{192} + 2^{96} - 1$ , we have, for example, that

$$2^{256} \equiv 2^{224} - 2^{192} - 2^{96} + 1 \pmod{p}.$$

Now, given

$$x \cdot y = z = \sum_{i=0}^{i<16} z_i \cdot 2^{32 \cdot i},$$

we can rewrite

$$z_8 \cdot 2^{256} \equiv z_8 \cdot 2^{224} - z_8 \cdot 2^{192} - z_8 \cdot 2^{96} + z_8 \cdot 1.$$

Keeping in mind that we compute

$$r = x \cdot y \pmod{p} = S_0 + 2S_1 + 2S_2 + S_3 + S_4 - S_5 - S_6 - S_7 - S_8 \pmod{p},$$

$z_8$  is identifiable at the right place(s) in

$S_0$	=	<	$z_0,$	$z_1,$	$z_2,$	$z_3,$	$z_4,$	$z_5,$	$z_6,$	$z_7$	>
$S_1$	=	<	0,	0,	0,	$z_{11},$	$z_{12},$	$z_{13},$	$z_{14},$	$z_{15}$	>
$S_2$	=	<	0,	0,	0,	$z_{12},$	$z_{13},$	$z_{14},$	$z_{15},$	0	>
$S_3$	=	<	$z_8,$	$z_9,$	$z_{10},$	0,	0,	0,	$z_{14},$	$z_{15}$	>
$S_4$	=	<	$z_9,$	$z_{10},$	$z_{11},$	$z_{13},$	$z_{14},$	$z_{15},$	$z_{13},$	$z_8$	>
$S_5$	=	<	$z_{11},$	$z_{12},$	$z_{13},$	0,	0,	0,	$z_8,$	$z_{10}$	>
$S_6$	=	<	$z_{12},$	$z_{13},$	$z_{14},$	$z_{15},$	0,	0,	$z_9,$	$z_{11}$	>
$S_7$	=	<	$z_{13},$	$z_{14},$	$z_{15},$	$z_8,$	$z_9,$	$z_{10},$	0,	$z_{12}$	>
$S_8$	=	<	$z_{14},$	$z_{15},$	0,	$z_9,$	$z_{10},$	$z_{11},$	0,	$z_{13}$	>

such that we add and/or subtract the right multiple(s).



## Part 2: in practice (6)

### Curve arithmetic

► **Concept:** for  $x, y \in \mathbb{F}_q$ , imagine we denote the efficiency of field operations as

$\mathcal{A}_{\mathbb{F}_q}$	$\models$	addition and subtraction	$\rightsquigarrow$	$x \pm y$	
$\mathcal{S}_{\mathbb{F}_q}$	$\models$	squaring	$\rightsquigarrow$	$x \cdot x$	$\equiv x^2$
$\mathcal{M}_{\mathbb{F}_q}$	$\models$	multiplication	$\rightsquigarrow$	$x \cdot y$	
$\mathcal{I}_{\mathbb{F}_q}$	$\models$	inversion	$\rightsquigarrow$	$1 / x$	$\equiv x^{-1}$

to support evaluation of the group operation.

## Part 2: in practice (7)

Curve arithmetic: projective representation

► **Idea:** somewhat *informally*,

$$\begin{array}{llll} \text{affine points} & \simeq & \text{2D points} & \Rightarrow P = (P_x, P_y) \in \mathbb{A}(K) \\ \text{projective points} & \simeq & \text{3D points} & \Rightarrow P = (P_x, P_y, P_z) \in \mathbb{P}(K) \end{array}$$

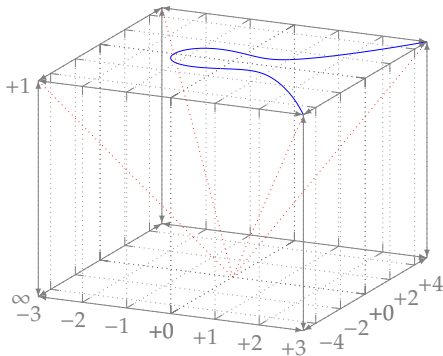
## Part 2: in practice (7)

### Curve arithmetic: projective representation

► **Idea:** somewhat *informally*,

$$\begin{array}{llll} \text{affine points} & \approx & \text{2D points} & \Rightarrow P = (P_x, P_y) \in \mathbb{A}(K) \\ \text{projective points} & \approx & \text{3D points} & \Rightarrow P = (P_x, P_y, P_z) \in \mathbb{P}(K) \end{array}$$

i.e., the latter means



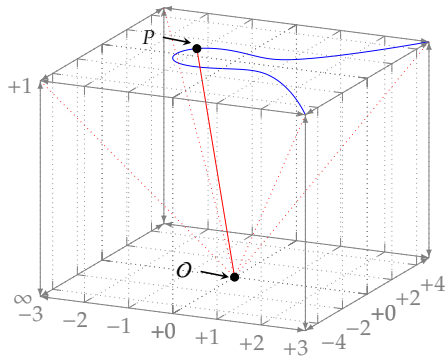
## Part 2: in practice (7)

### Curve arithmetic: projective representation

► **Idea:** somewhat *informally*,

$$\begin{array}{llll} \text{affine points} & \approx & \text{2D points} & \Rightarrow P = (P_x, P_y) \in \mathbb{A}(K) \\ \text{projective points} & \approx & \text{3D points} & \Rightarrow P = (P_x, P_y, P_z) \in \mathbb{P}(K) \end{array}$$

i.e., the latter means



## Part 2: in practice (7)

### Curve arithmetic: projective representation

► **Idea:** somewhat formally,

- Let  $K$  be a field, and  $c, d \in \mathbb{Z}$  such that  $c, d > 0$ .
- One can define an equivalence relation  $\sim$  on the set

$$K^3 \setminus \{(0, 0, 0)\}$$

by

$$(P_x, P_y, P_z) \sim (Q_x, Q_y, Q_z)$$

iff.

$$\begin{aligned} P_x &= \lambda^c Q_x \\ P_y &= \lambda^d Q_y \\ P_z &= \lambda Q_z \end{aligned}$$

for some  $\lambda \in K^*$ .

- The equivalence class containing

$$(x, y, z)$$

is

$$(x : y : z) = \{(\lambda^c x, \lambda^d y, \lambda z) \mid \lambda \in K^*\}$$

where  $(x, y, z)$  is a *representative* of the projective point  $(x : y : z)$ .

## Part 2: in practice (11)

### Curve arithmetic: projective representation

#### ► Idea: a comparison

	$\mathbb{A}(K)$ (affine)	$\mathbb{P}(K)$ (Jacobian projective)
Negation	$1\mathcal{A}_{\mathbb{F}_p}$	$1\mathcal{A}_{\mathbb{F}_p}$
Addition	$5\mathcal{A}_{\mathbb{F}_p} + 1\mathcal{S}_{\mathbb{F}_p} + 2\mathcal{M}_{\mathbb{F}_p} + 1\mathcal{I}_{\mathbb{F}_p}$	$7\mathcal{A}_{\mathbb{F}_p} + 4\mathcal{S}_{\mathbb{F}_p} + 12\mathcal{M}_{\mathbb{F}_p}$
Doubling	$8\mathcal{A}_{\mathbb{F}_p} + 2\mathcal{S}_{\mathbb{F}_p} + 2\mathcal{M}_{\mathbb{F}_p} + 1\mathcal{I}_{\mathbb{F}_p}$	$17\mathcal{A}_{\mathbb{F}_p} + 8\mathcal{S}_{\mathbb{F}_p} + 2\mathcal{M}_{\mathbb{F}_p}$
$\mathbb{A}(K) \mapsto \mathbb{P}(K)$ conversion		$1\mathcal{S}_{\mathbb{F}_p} + 3\mathcal{M}_{\mathbb{F}_p}$
$\mathbb{P}(K) \mapsto \mathbb{A}(K)$ conversion		$1\mathcal{S}_{\mathbb{F}_p} + 3\mathcal{M}_{\mathbb{F}_p} + 1\mathcal{I}_{\mathbb{F}_p}$

shows that

- we've basically traded less (i.e., no) inversions for more multiplications, meaning
- if (roughly)  $\mathcal{I}_{\mathbb{F}_p} > 10\mathcal{M}_{\mathbb{F}_p}$ , the Jacobian projective representation will be more efficient, *iff*.
- we minimise the number (and hence overhead) of conversions.

## Part 2: in practice (14)

### Curve parameters

- ▶ Idea: given

$$E : y^2 = x^3 + a_4x + a_6$$

one might select  $a_i$  to *optimise*, for example,

- ▶ the curve (and hence group) order, or
- ▶ point arithmetic.

## Part 2: in practice (14)

### Curve parameters

- ▶ Idea: given

$$E : y^2 = x^3 + a_4xz^4 + a_6z^6$$

one might select  $a_i$  to *optimise*, for example,

- ▶ the curve (and hence group) order, or
- ▶ point arithmetic.



► **Example:**

- In performing a Jacobian projective point doubling, we compute

$$3P_x^2 + P_z^4 a_4.$$

- By selecting  $a_4 = -3$ , we can calculate this term as

$$3(P_x - P_z^2)(P_x + P_z^2)$$

which is saves  $2S_{\mathbb{F}_p}$ .

## Part 2: in practice (15)

### Scalar multiplication

#### ► Idea:

1. we can rewrite and so reuse (multiplicative) exponentiation, i.e.,  $r = x^y$ , algorithms for (additive) exponentiation, i.e.,  $r = [y]x$ .

#### Algorithm (1MUL-L2R-BINARY)

**Input:** A group element  $x \in G^+$ , a base-2 integer  
 $0 \leq y < n$

**Output:** The group element  $r = [y]x \in G^+$

```
1  $r \leftarrow 0$ 
2 for  $i = |y| - 1$  downto 0 step -1 do
3    $r \leftarrow [2]r$ 
4   if  $y_i = 1$  then
5      $r \leftarrow r + x$ 
6   end
7 end
8 return  $r$ 
```

## Part 2: in practice (15)

### Scalar multiplication

#### ► Idea:

1. we can rewrite and so reuse (multiplicative) exponentiation, i.e.,  $r = x^y$ , algorithms for (additive) exponentiation, i.e.,  $r = [y]x$ .
2. we have, e.g., that

for a	$x \in \mathbb{Z}_N^\times$	computing	$1/x \pmod{N}$	is relatively	expensive
for a	$P \in E(\mathbb{F}_q)$	computing	$-P$	is relatively	<i>inexpensive</i>

meaning we can capitalise on a *signed* representation of  $y$ .

## Part 2: in practice (15)

### Scalar multiplication

#### ► Idea:

1. we can rewrite and so reuse (multiplicative) exponentiation, i.e.,  $r = x^y$ , algorithms for (additive) exponentiation, i.e.,  $r = [y]x$ .
2. we have, e.g., that

$$\begin{array}{ll} \text{for a } x \in \mathbb{Z}_N^\times & \text{computing } 1/x \pmod{N} \text{ is relatively expensive} \\ \text{for a } P \in E(\mathbb{F}_q) & \text{computing } -P \text{ is relatively inexpensive} \end{array}$$

meaning we can capitalise on a *signed* representation of  $y$ , e.g.,

### Definition

A **Non-Adjacent Form (NAF)** of some positive integer  $y$  is

$$\begin{aligned} \hat{y} &= \langle \hat{y}_0, \hat{y}_1, \dots, \hat{y}_{n-1} \rangle \\ &\mapsto y \\ &= \sum_{i=0}^{n-1} \hat{y}_i \cdot 2^i \end{aligned}$$

such that  $\hat{y}_i \in \{-1, 0, +1\}$ , and two specific conditions hold:

1. the most-significant digit of  $\hat{y}$  is non-zero, i.e.,  $\hat{y}_{n-1} \neq 0$
2. no two consecutive digits in  $\hat{y}$  are non-zero, i.e., if  $\hat{y}_i \neq 0$  then either  $\hat{y}_{i+1} = 0$  and/or  $\hat{y}_{i-1} = 0$ .

## Part 2: in practice (15)

### Scalar multiplication

#### ► Idea:

1. we can rewrite and so reuse (multiplicative) exponentiation, i.e.,  $r = x^y$ , algorithms for (additive) exponentiation, i.e.,  $r = [y]x$ .
2. we have, e.g., that

for a  $x \in \mathbb{Z}_N^\times$  computing  $1/x \pmod{N}$  is relatively expensive  
for a  $P \in E(\mathbb{F}_q)$  computing  $-P$  is relatively *inexpensive*

meaning we can capitalise on a *signed* representation of  $y$ , e.g.,

#### Algorithm (RECODE-NAF)

**Input:** An integer  $y$

**Output:** A sequence  $y'$  which is the NAF representation of  $y$

```
1  $y' \leftarrow \emptyset, i \leftarrow 0$ 
2 while  $y \geq 1$  do
3   if  $y \equiv 1 \pmod{2}$  then
4      $| y'_i \leftarrow 2 - (y \bmod 4), y \leftarrow y - y'_i$ 
5   else
6      $| y'_i \leftarrow 0$ 
7   end
8    $y \leftarrow \lfloor y/2 \rfloor, i \leftarrow i + 1$ 
9 end
10 return  $y'$ 
```

#### Algorithm (1MUL-L2R-NAF)

**Input:** A group element  $x \in G^+$ , a base-2 integer  $0 \leq y < n$

**Output:** The group element  $r = [y]x \in G^+$

```
1  $y' \leftarrow \text{RECODE-NAF}(y)$ 
2  $r \leftarrow 0$ 
3 for  $i = |y'| - 1$  downto 0 step -1 do
4    $r \leftarrow [2]r$ 
5   if  $y'_i = +1$  then
6      $| r \leftarrow r + x$ 
7   else if  $y'_i = -1$  then
8      $| r \leftarrow r - x$ 
9   end
10 end
11 return  $r$ 
```

- ▶ **Take away points:** you can often simply *use*

$$Q = [k]P \in E(\mathbb{F}_q),$$

but understanding internals of this primitive can be useful and/or important.

- ▶ some historically interesting aspects; some “portable” concepts,
- ▶ close relationship between primitive and underlying Mathematics,
- ▶ wide range of viable implementation strategies,
- ▶ extensive deployment, in various contexts and use-cases.

## References

- [1] D.J. Bernstein and T. Lange. *ECCHacks*. URL: <http://ecchacks.cr.yp.to> (see pp. 2–16).
- [2] D.J. Bernstein and T. Lange. *Explicit-Formulas Database (EFD)*. URL: <http://www.hyperelliptic.org/EFD>.
- [3] J. Renes, C. Costello, and L. Batina. “Complete addition formulas for prime order elliptic curves”. In: *Advances in Cryptology (EUROCRYPT)*. LNCS 9665. Springer-Verlag, 2016, pp. 403–428.
- [4] J.A. Solinas. *Generalized Mersenne Numbers*. Tech. rep. CORR 99-39. Centre for Applied Cryptographic Research (CACR), University of Waterloo, 1999 (see p. 39).
- [5] *Digital Signature Standard (DSS)*. National Institute of Standards and Technology (NIST) Federal Information Processing Standard (FIPS) 186-4. 2013. URL: <http://csrc.nist.gov> (see pp. 34–37).
- [6] N.P. Smart, ed. *Algorithms, key size and parameters report*. European Union Agency for Network and Information Security (ENISA). 2014. URL: <http://www.enisa.europa.eu> (see p. 38).