Applied Cryptology

Daniel Page

Department of Computer Science,
University Of Bristol,
Merchant Venturers Building,
Woodland Road,
Bristol, BS8 1UB. UK.
⟨csdsp@bristol.ac.uk⟩

April 24, 2024

Keep in mind there are *two* PDFs available (of which this is the latter):

1. a PDF of examinable material used as lecture slides, and
2. a PDF of non-examinable, extra material:
   - the associated notes page may be pre-populated with extra, written explaination of material covered in lecture(s), plus
   - anything with a "grey'ed out" header/footer represents extra material which is useful and/or interesting but out of scope (and hence not covered).

Notes:

Notes:

- Agenda: explore **elliptic curves** $\rightsquigarrow$ **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!

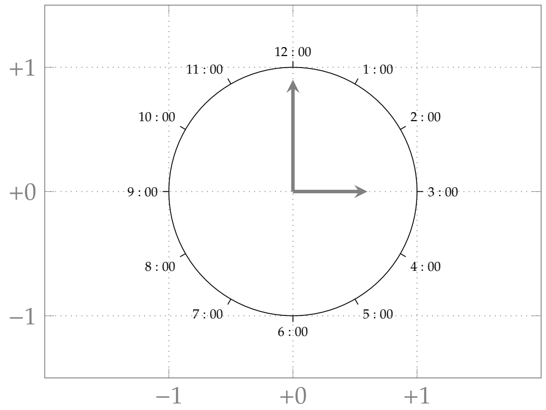
  $\sim$ 2 hours $\Rightarrow$ introductory, and (very) selective (versus definitive) coverage.

Notes:

Part 1: in theory (1)

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

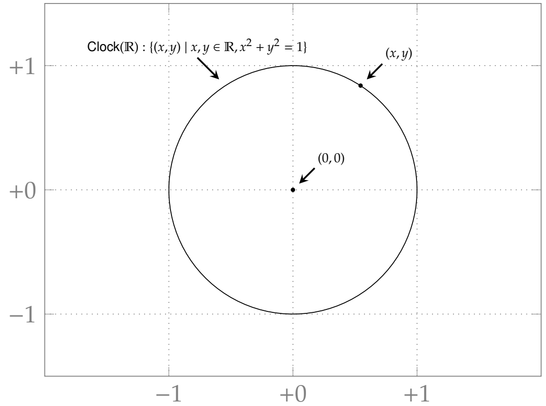
- Idea: consider a clock face.



Notes:

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)



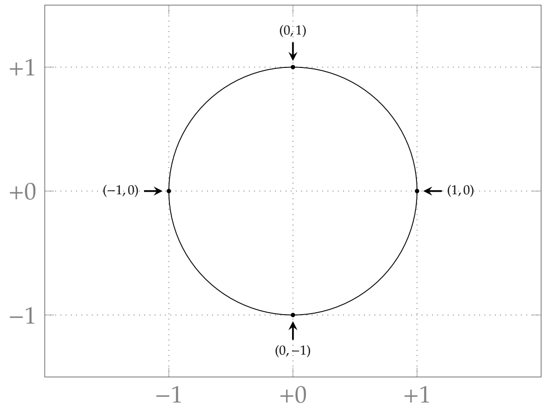
where we can describe points in Cartesian form, namely

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

Notes:

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)

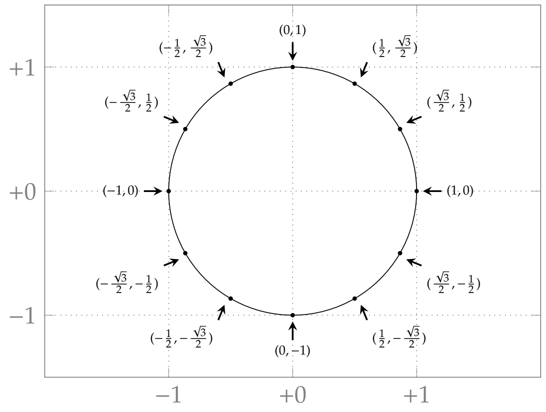


where we can describe points in Cartesian form, namely

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

Notes:

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



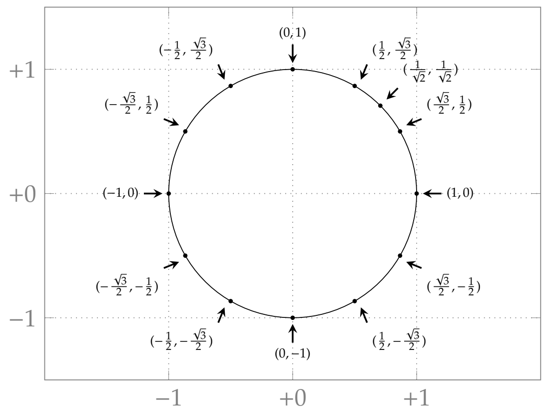
where we can describe points in Cartesian form, namely

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

Notes:

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)

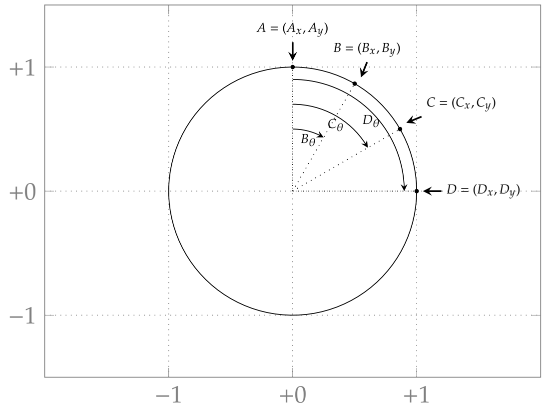


where we can describe points in Cartesian form, namely

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

Notes:

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



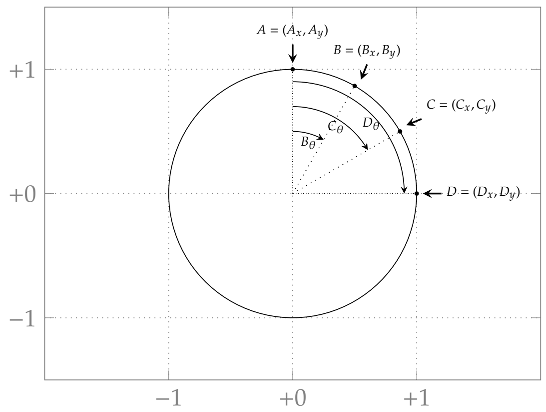
where we can describe points in Cartesian *or* parametric form, namely

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

Notes:

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

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

Notes:

© Daniel Page (csdsp@bristol.ac.uk)
Applied Cryptology
University of BRISTOL
git # c8178615 @ 2024-04-24

Theorem

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

- "Proof":

P \ 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

Notes:

Part 1: in theory (2)
An ECCHacks-based primer [1]: "from geometry to group theory"

Theorem

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

- "Proof":

P \ 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:

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

Notes:

Part 1: in theory (2)

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

Theorem

The set of "o'clock points" $\mathsf{O'Clock}(\mathbb{R}) \subset \mathsf{Clock}(\mathbb{R})$ 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
P 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:

$$
\begin{array}{rcl}
\text{"1:00"} \oplus \text{"12:00"} & = & (\sin 30, \cos 30) \oplus (\sin 0, \cos 0) \\
 & = & (\sin(30+0), \cos(30+0)) \\
 & = & (\sin 30, \cos 30) \\
 & = & \text{"1:00"}
\end{array}
$$

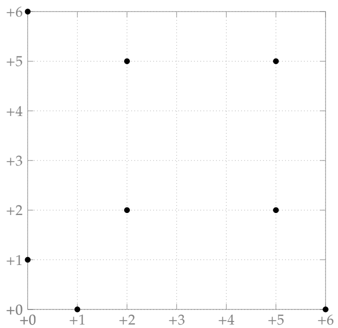
Notes:

Part 1: in theory (3)

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

- Idea: eliminate non-discrete aspect of $\mathsf{Clock}(\mathbb{R})$, by
  1. considering
  $$\mathsf{Clock}(\mathbb{F}_q) : \{(x, y) \mid x, y \in \mathbb{F}_q, x^2 + y^2 = 1\},$$
  then
  2. translating
  $$\begin{array}{rcl} R = (R_x, R_y) = P \oplus Q & = & (\sin P_\theta \cos Q_\theta + \cos P_\theta \sin Q_\theta, \cos P_\theta \cos Q_\theta - \sin P_\theta \sin Q_\theta) \\ & \equiv & (P_x Q_y + P_y Q_x, P_y Q_y - P_x Q_x) \end{array}$$

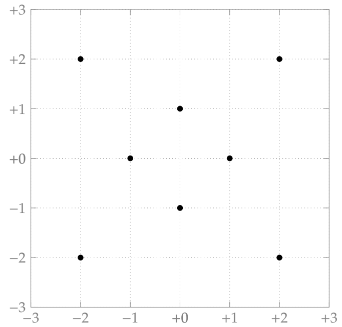
  such that if $q = 7$, for example, we *naturally* get a set of discrete points:



Notes:

- Idea: eliminate non-discrete aspect of $\mathsf{Clock}(\mathbb{R})$, by
  1. considering
     $$\mathsf{Clock}(\mathbb{F}_q) : \{(x, y) \mid x, y \in \mathbb{F}_q, x^2 + y^2 = 1\},$$
     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:



Notes:

Theorem

The set $\mathsf{Clock}(\mathbb{F}_7)$ 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)

Notes:

Part 1: in theory (4)
An ECCHacks-based primer [1]: "from geometry to group theory"

Theorem

The set $\mathsf{Clock}(\mathbb{F}_7)$ 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:

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

Notes:

Part 1: in theory (4)
An ECCHacks-based primer [1]: "from geometry to group theory"

Theorem

The set $\mathsf{Clock}(\mathbb{F}_7)$ 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:

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

Notes:

Part 1: in theory (5)

General elliptic curves over K

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

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

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

Notes:

Part 1: in theory (6)

General elliptic curves over K

Definition

For a given curve E, if

$$\begin{array}{rcl}
b_2 & = & {a_1}^2 + 4a_2 \\
b_4 & = & a_1a_3 + 2a_4 \\
b_6 & = & {a_3}^2 + 4a_6 \\
b_8 & = & {a_1}^2a_6 + 4a_2a_6 - a_1a_3a_4 + a_2{a_3}^2 - {a_4}^2 \\
c_4 & = & b_2^2 - 24b_4 \\
c_6 & = & -b_2^3 + 36b_2b_4 - 216b_6
\end{array}$$

then we can define

1. the **discriminant** of E as

$$\Delta = -b_2^2b_8 - 8b_4^3 - 27b_6^2 + 9b_2b_4b_6$$

   and

2. the **j-invariant** of E as

$$j(E) = c_4^3/\Delta.$$

- Translation:
  - The discriminant roughly tells us the *shape* of the curve; we can avoid **singular** curves, i.e., curves with "cusps", by avoiding those where $\Delta = 0$.
  - The j-invariant roughly tells us the *family* of the curve; if for curves E and E' over K we have $j(E) = j(E')$, then the curves are **isomorphic**.

Notes:

Definition

Given

$$P = (P_x, P_y) \in E(K)$$

and

$$P' = (P'_x, P'_y) \in E'(K)$$

then E and E' are **isomorphic** iff. there exist constants $r, s, t \in K$ and $u \in K^\times$ such that

$$\begin{array}{rcl} P_x & = & u^2 P'_x + r \\ P_y & = & u^3 P'_y + s u^2 P'_x + t \end{array}$$

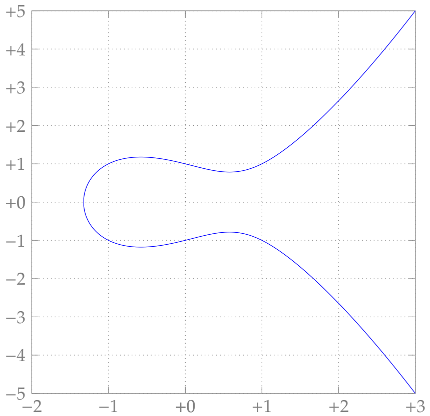
maps P' into P, i.e, transforms E' into E (and vice versa).

- Translation:
  - If two curves E and E' over K are isomorphic, a bi-rational mapping exists between them (which preserves the point at infinity).
  - This is much less complicated than it looks: the mapping is effectively just an admissible (i.e., sane) change of variables.

Notes:

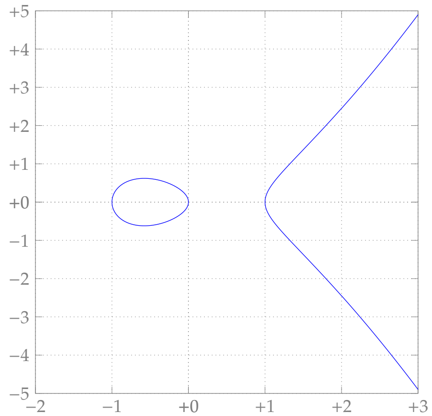
Example

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 $j(E) = 1728$:



Notes:

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

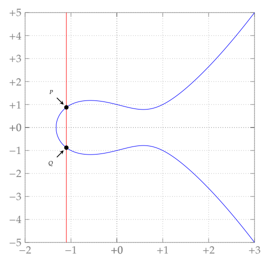
$$P \oplus Q \oplus T = \mathcal{O}$$

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

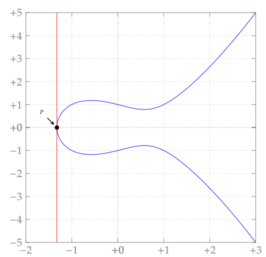
- Idea: the **chord-tangent** (or **line-tangent**) process affords a group operation:
  - Consider $P, Q \in E(K)$:
    - if $P \neq Q$ then a line drawn between them will intersect E at T,
    - if $P = Q$ then a line drawn between them is a tangent to E.
  - The group operation is then just geometry:
    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.

Notes:

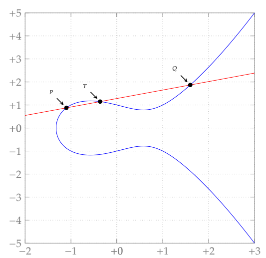
Case #1: $P \oplus Q \oplus \mathcal{O} = \mathcal{O} \Rightarrow P = -Q$



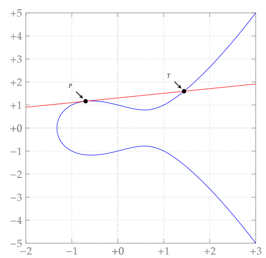
Case #2: $P \oplus P \oplus \mathcal{O} = \mathcal{O}$



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

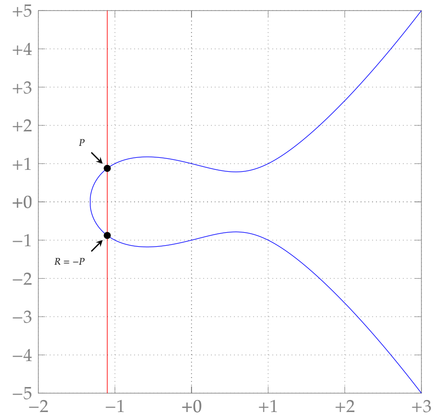


Case #4: $P \oplus P \oplus T = \mathcal{O} \Rightarrow T = -(P \oplus P)$



Notes:

Algorithm (affine point negation)

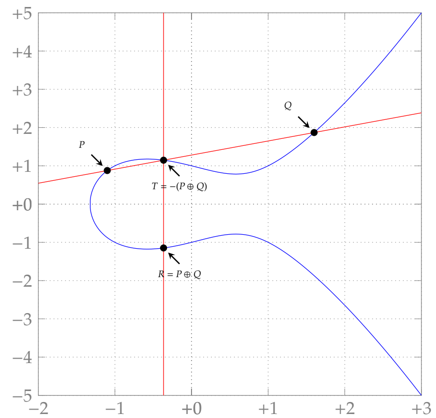


- For $P = (P_x, P_y)$, $R = (R_x, R_y) = -P$ is computed via

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

Notes:

Algorithm (affine point addition)



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

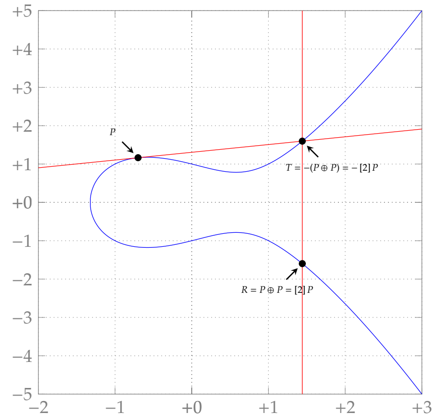
$$\begin{array}{rcl} \lambda & = & \frac{Q_y - P_y}{Q_x - P_x} \\ \mu & = & \frac{P_y Q_x - Q_y P_x}{Q_x - P_x} \end{array}$$

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

$$\begin{array}{rcl} R_x & = & \lambda^2 + a_1 \lambda - a_2 - P_x - Q_x \\ R_y & = & -(\lambda + a_1) R_x - \mu - a_3 \end{array}$$

Notes:

Algorithm (affine point doubling)



- For $P = (P_x, P_y)$, let

$$\begin{array}{rcl} \lambda & = & \frac{3P_x^2+2a_2P_x+a_4-a_1P_y}{2P_y+a_1P_x+a_3} \\ \mu & = & \frac{-P_x^3+a_4P_x+2a_6-a_3P_y}{2P_y+a_1P_x+a_3} \end{array}$$

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

$$\begin{array}{rcl} R_x & = & \lambda^2 + a_1\lambda - a_2 - P_x - Q_x \\ R_y & = & -(\lambda + a_1)R_x - \mu - a_3 \end{array}$$

Notes:

- 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 $\mathbb{F}_{2^m}$

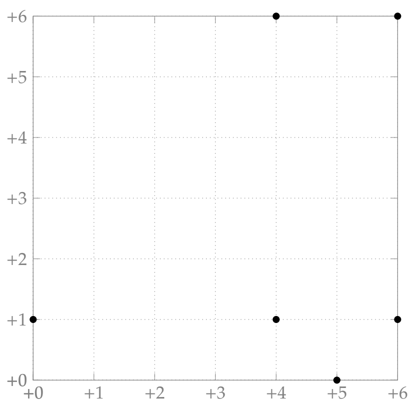
meaning we can

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.

Notes:

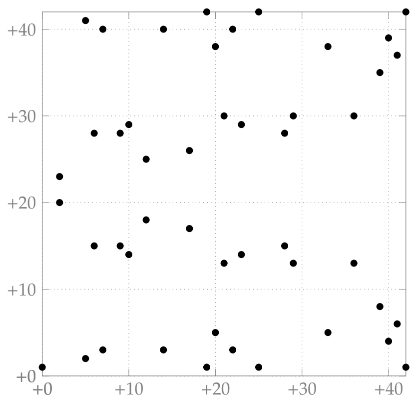
Example

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

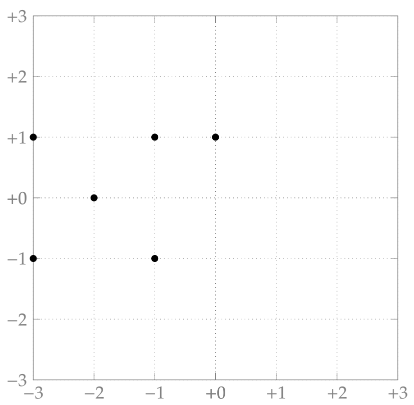
Consider $E(\mathbb{F}_{43})$ for $a_1 = 0$, $a_2 = 0$, $a_3 = 0$, $a_4 = 1$, and $a_6 = 3$, meaning $\Delta = 4$ and $j(E) = 1$:



Notes:

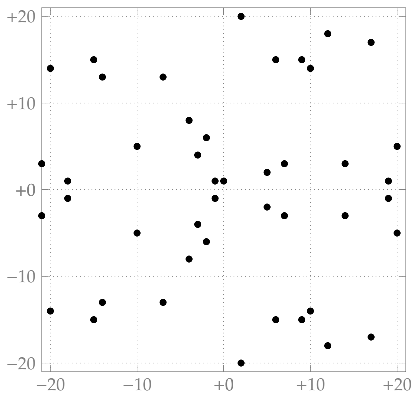
Example

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 $j(E) = 1$:



Notes:

Algorithm (specialisation for $q = p$)

1. Specialise the Weierstraß equation to
$$E : y^2 = x^3 + a_4 x + a_6.$$
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
$$\begin{aligned} R_x &= \lambda^2 - P_x - Q_x \\ R_y &= \lambda(P_x - R_x) - P_y \end{aligned}$$

Notes:

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

1. Specialise the Weierstraß equation to
$$E : y^2 + xy = x^3 + a_2 x^2 + a_6.$$
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
$$\lambda = \begin{cases} \frac{Q_y + P_y}{Q_x + P_x} & \text{if } P \neq Q \\ \frac{P_x^2 + P_y}{P_x} & \text{if } P = Q \end{cases}$$
then
$$\begin{aligned} R_x &= \lambda^2 + \lambda + a_2 + P_x + Q_x \\ R_y &= \lambda(P_x + R_x) + R_x + P_y \end{aligned}$$

Notes:

Part 1: in theory (17)

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

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

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

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

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

and in fact we can describe the entire group operation as

		Q					
		$\mathcal{O}$	(4,1)	(4,6)	(5,0)	(6,1)	(6,6)
	$\mathcal{O}$	$\mathcal{O}$	(4,1)	(4,6)	(5,0)	(6,1)	(6,6)
	(4,1)	(4,1)	(6,6)	$\mathcal{O}$	(6,1)	(4,6)	(5,0)
	(4,6)	(4,6)	$\mathcal{O}$	(6,1)	(6,6)	(5,0)	(4,1)
P	(5,0)	(5,0)	(6,1)	(6,6)	$\mathcal{O}$	(4,1)	(4,6)
	(6,1)	(6,1)	(4,6)	(5,0)	(4,1)	(6,6)	$\mathcal{O}$
	(6,6)	(6,6)	(5,0)	(4,1)	(4,6)	$\mathcal{O}$	(6,1)

Notes:

Part 1: in theory (18)

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

Example

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

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

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

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

and in fact we can describe the entire group operation as

		Q							
		$\mathcal{O}$	(0,1)	(1,0)	(1,1)	(x,0)	(x,x)	(x+1,0)	(x+1,x+1)
	$\mathcal{O}$	$\mathcal{O}$	(0,1)	(1,0)	(1,1)	(x,0)	(x,x)	(x+1,0)	(x+1,x+1)
	(0,1)	(0,1)	$\mathcal{O}$	(1,1)	(1,0)	(x+1,0)	(x+1,x+1)	(x,0)	(x,x)
	(1,0)	(1,0)	(1,1)	(0,1)	$\mathcal{O}$	(x+1,x+1)	(x,0)	(x,x)	(x+1,0)
P	(1,1)	(1,1)	(1,0)	$\mathcal{O}$	(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)	$\mathcal{O}$	(1,1)	(0,1)
	(x,x)	(x,x)	(x+1,x+1)	(x,0)	(x+1,0)	$\mathcal{O}$	(1,1)	(0,1)	(1,0)
	(x+1,0)	(x+1,0)	(x,0)	(x,x)	(x+1,x+1)	(1,1)	(0,1)	(1,0)	$\mathcal{O}$
	(x+1,x+1)	(x+1,x+1)	(x,x)	(x+1,0)	(x,0)	(0,1)	(1,0)	$\mathcal{O}$	(1,1)

Notes:

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

$$Q = [l]\,P,$$

$l \in \{0, 1, \ldots, n-1\}$ 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 $l = \log_P Q$.

Notes:

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

$$Q = [l]\,P,$$

$l \in \{0, 1, \ldots, n-1\}$ 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 $\mathbb{G}^{+} = (E(K), +)$,
  2. we can define **scalar multiplication** as

     $$Q = [l]\,P$$

     i.e.,

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

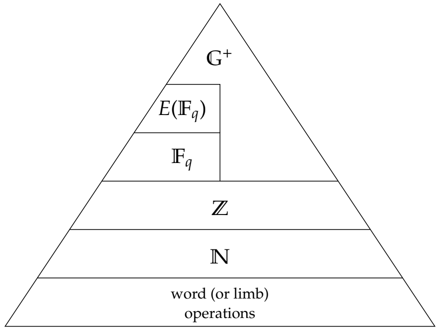
     and
  3. doing so allows us to pose a (EC-)DLP such that
     - given P and l, it is *easy* to compute Q, *but*
     - given P and Q, it is *hard* to compute l (on a suitable curve)

     and thus use it in a group-based scheme.

Notes:

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, ...

      ⋮

Notes:

Part 2: in practice (2)

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

```
DSA.ParamGen begin
    generate a prime q
    generate a prime p such that q | p − 1
    generate a G^× = F_q = ⟨g⟩ of order q
end
```

```
EC-DSA.ParamGen begin


    generate a G^+ = E(F_q) = ⟨G⟩ of order n
end
```

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

Notes:

Part 2: in practice (2)

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

```
DSA.KeyGen begin
  select x <-$- {1, 2, ..., q − 1}
  compute y = g^x (mod p)
  return (x, y)
end
```

```
EC-DSA.KeyGen begin
  select l <-$- {1, 2, ..., n − 1}
  compute Q = [l] G
  return (l, Q)
end
```

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

Notes:

Part 2: in practice (2)

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

DSA.Sign begin
- select $k \xleftarrow{\$} \{1, 2, \ldots, q-1\}$
- compute $h = H(m)$
- compute $r = (g^k \bmod p) \bmod q$
- compute $s = (k^{-1} \cdot (h + r \cdot x)) \bmod q$
- return (r, s)

end

EC-DSA.Sign begin
- select $k \xleftarrow{\$} \{1, 2, \ldots, n-1\}$
- compute $h = \mathrm{LSB}_{|n|}(H(m))$
- compute $r = R_x \bmod n$ where $(R_x, R_y) = R = [k]\,G$
- compute $s = (k^{-1} \cdot (h + r \cdot l)) \bmod n$
- return (r, s)

end

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

Notes:

© Daniel Page ⟨csdsp@bristol.ac.uk⟩
Applied Cryptology
University of BRISTOL
git # c8178615 @ 2024-04-24

Part 2: in practice (2)

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

DSA.Verify begin
- return **false** if $0 \not< r, s$ or $r, s \not< n$
- compute $h = H(m)$
- compute $u_1 = h \cdot s^{-1} \bmod q$
- compute $u_2 = r \cdot s^{-1} \bmod q$
- compute $v = (g^{u_1} \cdot y^{u_2} \bmod p) \bmod q$
- return **false** if $r \neq v$, else return **true**

end

EC-DSA.Verify begin
- return **false** if $0 \not< r, s$ or $r, s \not< n$
- compute $h = \mathrm{LSB}_{|n|}(H(m))$
- compute $u_1 = h \cdot s^{-1} \bmod n$
- compute $u_2 = r \cdot s^{-1} \bmod n$
- compute $v = V_x \bmod n$ where $(V_x, V_y) = V = [u_1]G + [u_2]Q$
- return **false** if $r \neq v$, else return **true**

end

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

Notes:

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.

Notes:

Part 2: in practice (4)

Field parameters

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

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

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 $0 \leq r < p$.

Notes:

Part 2: in practice (5)

Field parameters

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

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

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

Notes:

Part 2: in practice (6)

Curve arithmetic

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

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

  to support evaluation of the group operation.

Notes:

Part 2: in practice (7)

Curve arithmetic: projective representation

- Idea: somewhat *in*formally,

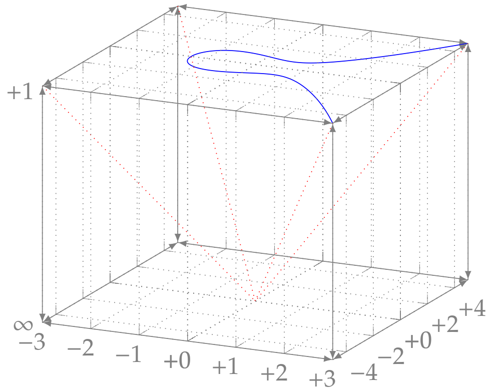
$$
\begin{array}{rclcrcl}
\textbf{affine points} & \simeq & \text{2D points} & \Rightarrow & P = (P_x, P_y) & \in & \mathbb{A}(K) \\
\textbf{projective points} & \simeq & \text{3D points} & \Rightarrow & P = (P_x, P_y, P_z) & \in & \mathbb{P}(K)
\end{array}
$$

Notes:

- Idea: somewhat *in*formally,

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

  i.e., the latter means



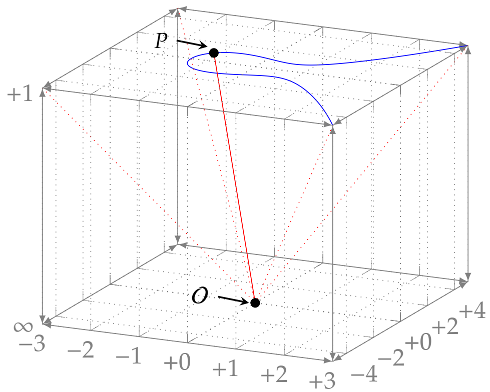
Notes:

Part 2: in practice (7)
Curve arithmetic: projective representation

- Idea: somewhat *in*formally,

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

  i.e., the latter means



Notes:

- 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.

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

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 $(x : y : z)$.

Notes:

Definition

Let $K = \mathbb{F}_p$. The so-called **Jacobian projective representation** for points on $E(\mathbb{F}_p)$ sets $c = 2$ and $d = 3$, then alters the Weierstraß equation

$$E : y^2 = x^3 + a_4 x z^4 + a_6 z^6.$$

This means the K-rational set of points on E is now

$$E(\mathbb{F}_p) : \{(x, y, z) \mid x, y, z \in \mathbb{F}_p, y^2 = x^3 + a_4 x z^4 + a_6 z^6\} \cup \{\mathcal{O}\},$$

with $\mathcal{O} = (1 : 1 : 0)$ and

$$(x, y) \in \mathbb{A}(\mathbb{F}_p) \longleftrightarrow (\lambda^2 x, \lambda^3 y, \lambda) \in \mathbb{P}(\mathbb{F}_p)$$

Notes:

Algorithm (affine addition)

$$
\begin{array}{lll}
\lambda_1 & \leftarrow Q_y - P_y & 1\mathcal{A}_{\mathbb{F}_p} \\
\lambda_2 & \leftarrow Q_x - P_x & 1\mathcal{A}_{\mathbb{F}_p} \\
\lambda_3 & \leftarrow \lambda_2^{-1} & 1\mathcal{I}_{\mathbb{F}_p} \\
\lambda_4 & \leftarrow \lambda_1 \cdot \lambda_3 & 1\mathcal{M}_{\mathbb{F}_p} \\
\lambda_5 & \leftarrow \lambda_4^2 & 1\mathcal{S}_{\mathbb{F}_p} \\
\lambda_6 & \leftarrow \lambda_5 - P_x & 1\mathcal{A}_{\mathbb{F}_p} \\
R_x & \leftarrow \lambda_6 - Q_x & 1\mathcal{A}_{\mathbb{F}_p} \\
\lambda_7 & \leftarrow P_x - R_x & 1\mathcal{A}_{\mathbb{F}_p} \\
\lambda_8 & \leftarrow \lambda_4 \cdot \lambda_7 & 1\mathcal{M}_{\mathbb{F}_p} \\
R_y & \leftarrow \lambda_8 - P_y & 1\mathcal{A}_{\mathbb{F}_p} \\
\hline
 & & 5\mathcal{A}_{\mathbb{F}_p} + 1\mathcal{S}_{\mathbb{F}_p} + 2\mathcal{M}_{\mathbb{F}_p} + 1\mathcal{I}_{\mathbb{F}_p}
\end{array}
$$

Algorithm (affine doubling)

$$
\begin{array}{lll}
\lambda_1 & \leftarrow P_x^2 & 1\mathcal{S}_{\mathbb{F}_p} \\
\lambda_2 & \leftarrow \lambda_1 + \lambda_1 & 1\mathcal{A}_{\mathbb{F}_p} \\
\lambda_3 & \leftarrow \lambda_1 + \lambda_2 & 1\mathcal{A}_{\mathbb{F}_p} \\
\lambda_4 & \leftarrow \lambda_3 + a_4 & 1\mathcal{A}_{\mathbb{F}_p} \\
\lambda_5 & \leftarrow P_y + P_y & 1\mathcal{A}_{\mathbb{F}_p} \\
\lambda_6 & \leftarrow \lambda_5^{-1} & 1\mathcal{I}_{\mathbb{F}_p} \\
\lambda_7 & \leftarrow \lambda_4 \cdot \lambda_6 & 1\mathcal{M}_{\mathbb{F}_p} \\
\lambda_8 & \leftarrow \lambda_7^2 & 1\mathcal{S}_{\mathbb{F}_p} \\
\lambda_9 & \leftarrow \lambda_8 - P_x & 1\mathcal{A}_{\mathbb{F}_p} \\
R_x & \leftarrow \lambda_9 - Q_x & 1\mathcal{A}_{\mathbb{F}_p} \\
\lambda_{10} & \leftarrow P_x - R_x & 1\mathcal{A}_{\mathbb{F}_p} \\
\lambda_{11} & \leftarrow \lambda_7 \cdot \lambda_{10} & 1\mathcal{M}_{\mathbb{F}_p} \\
R_y & \leftarrow \lambda_{11} - P_y & 1\mathcal{A}_{\mathbb{F}_p} \\
\hline
 & & 8\mathcal{A}_{\mathbb{F}_p} + 2\mathcal{S}_{\mathbb{F}_p} + 2\mathcal{M}_{\mathbb{F}_p} + 1\mathcal{I}_{\mathbb{F}_p}
\end{array}
$$

Notes:

Algorithm (projective addition [2])

$$
\begin{array}{lll}
\lambda_1 & \leftarrow P_z^2 & 1\mathcal{S}_{\mathbb{F}_p} \\
\lambda_2 & \leftarrow P_z \cdot \lambda_1 & 1\mathcal{M}_{\mathbb{F}_p} \\
\lambda_3 & \leftarrow Q_z^2 & 1\mathcal{S}_{\mathbb{F}_p} \\
\lambda_4 & \leftarrow Q_z \cdot \lambda_3 & 1\mathcal{M}_{\mathbb{F}_p} \\
\lambda_5 & \leftarrow P_x \cdot \lambda_3 & 1\mathcal{M}_{\mathbb{F}_p} \\
\lambda_6 & \leftarrow Q_x \cdot \lambda_1 & 1\mathcal{M}_{\mathbb{F}_p} \\
\lambda_7 & \leftarrow \lambda_6 - \lambda_5 & 1\mathcal{A}_{\mathbb{F}_p} \\
\lambda_8 & \leftarrow P_y \cdot \lambda_4 & 1\mathcal{M}_{\mathbb{F}_p} \\
\lambda_9 & \leftarrow Q_y \cdot \lambda_2 & 1\mathcal{M}_{\mathbb{F}_p} \\
\lambda_{10} & \leftarrow \lambda_9 - \lambda_8 & 1\mathcal{A}_{\mathbb{F}_p} \\
\lambda_{11} & \leftarrow \lambda_7^2 & 1\mathcal{S}_{\mathbb{F}_p} \\
\lambda_{12} & \leftarrow \lambda_7 \cdot \lambda_{11} & 1\mathcal{M}_{\mathbb{F}_p} \\
\lambda_{13} & \leftarrow \lambda_5 \cdot \lambda_{11} & 1\mathcal{M}_{\mathbb{F}_p} \\
\lambda_{14} & \leftarrow P_z \cdot Q_z & 1\mathcal{M}_{\mathbb{F}_p} \\
\lambda_{15} & \leftarrow \lambda_{13} + \lambda_{13} & 1\mathcal{A}_{\mathbb{F}_p} \\
\lambda_{16} & \leftarrow \lambda_{15} + \lambda_{12} & 1\mathcal{A}_{\mathbb{F}_p} \\
\lambda_{17} & \leftarrow \lambda_{10}^2 & 1\mathcal{S}_{\mathbb{F}_p} \\
R_x & \leftarrow \lambda_{17} - \lambda_{16} & 1\mathcal{A}_{\mathbb{F}_p} \\
R_z & \leftarrow \lambda_7 \cdot \lambda_{14} & 1\mathcal{M}_{\mathbb{F}_p} \\
\lambda_{18} & \leftarrow \lambda_{13} - R_x & 1\mathcal{A}_{\mathbb{F}_p} \\
\lambda_{19} & \leftarrow \lambda_{10} \cdot \lambda_{18} & 1\mathcal{M}_{\mathbb{F}_p} \\
\lambda_{20} & \leftarrow \lambda_8 \cdot \lambda_{12} & 1\mathcal{M}_{\mathbb{F}_p} \\
R_y & \leftarrow \lambda_{19} - \lambda_{20} & 1\mathcal{A}_{\mathbb{F}_p} \\
\hline
 & & 7\mathcal{A}_{\mathbb{F}_p} + 4\mathcal{S}_{\mathbb{F}_p} + 12\mathcal{M}_{\mathbb{F}_p} + 0\mathcal{I}_{\mathbb{F}_p}
\end{array}
$$

Algorithm (projective doubling [2])

$$
\begin{array}{lll}
\lambda_1 & \leftarrow P_x^2 & 1\mathcal{S}_{\mathbb{F}_p} \\
\lambda_2 & \leftarrow P_y^2 & 1\mathcal{S}_{\mathbb{F}_p} \\
\lambda_4 & \leftarrow P_z^2 & 1\mathcal{S}_{\mathbb{F}_p} \\
\lambda_3 & \leftarrow \lambda_2^2 & 1\mathcal{S}_{\mathbb{F}_p} \\
\lambda_5 & \leftarrow \lambda_2 + P_x & 1\mathcal{A}_{\mathbb{F}_p} \\
\lambda_6 & \leftarrow \lambda_5^2 & 1\mathcal{S}_{\mathbb{F}_p} \\
\lambda_7 & \leftarrow \lambda_6 - \lambda_1 & 1\mathcal{A}_{\mathbb{F}_p} \\
\lambda_8 & \leftarrow \lambda_7 - \lambda_3 & 1\mathcal{A}_{\mathbb{F}_p} \\
\lambda_9 & \leftarrow \lambda_8 + \lambda_8 & 1\mathcal{A}_{\mathbb{F}_p} \\
\lambda_{10} & \leftarrow \lambda_4^2 & 1\mathcal{S}_{\mathbb{F}_p} \\
\lambda_{11} & \leftarrow \lambda_{10} \cdot a_4 & 1\mathcal{M}_{\mathbb{F}_p} \\
\lambda_{12} & \leftarrow \lambda_1 + \lambda_1 & 1\mathcal{A}_{\mathbb{F}_p} \\
\lambda_{13} & \leftarrow \lambda_{12} + \lambda_1 & 1\mathcal{A}_{\mathbb{F}_p} \\
\lambda_{14} & \leftarrow \lambda_{13} + \lambda_{11} & 1\mathcal{A}_{\mathbb{F}_p} \\
\lambda_{15} & \leftarrow \lambda_{14}^2 & 1\mathcal{S}_{\mathbb{F}_p} \\
\lambda_{16} & \leftarrow \lambda_9 + \lambda_9 & 1\mathcal{A}_{\mathbb{F}_p} \\
R_x & \leftarrow \lambda_{15} - \lambda_{16} & 1\mathcal{A}_{\mathbb{F}_p} \\
\lambda_{17} & \leftarrow \lambda_9 - R_x & 1\mathcal{A}_{\mathbb{F}_p} \\
\lambda_{18} & \leftarrow \lambda_3 + \lambda_3 & 1\mathcal{A}_{\mathbb{F}_p} \\
\lambda_{19} & \leftarrow \lambda_{18} + \lambda_{18} & 1\mathcal{A}_{\mathbb{F}_p} \\
\lambda_{20} & \leftarrow \lambda_{19} + \lambda_{19} & 1\mathcal{A}_{\mathbb{F}_p} \\
\lambda_{21} & \leftarrow \lambda_{14} \cdot \lambda_{17} & 1\mathcal{M}_{\mathbb{F}_p} \\
R_y & \leftarrow \lambda_{21} - \lambda_{20} & 1\mathcal{A}_{\mathbb{F}_p} \\
\lambda_{22} & \leftarrow P_y + P_z & 1\mathcal{A}_{\mathbb{F}_p} \\
\lambda_{23} & \leftarrow \lambda_{22}^2 & 1\mathcal{S}_{\mathbb{F}_p} \\
\lambda_{24} & \leftarrow \lambda_{23} - \lambda_2 & 1\mathcal{A}_{\mathbb{F}_p} \\
R_z & \leftarrow \lambda_{24} - \lambda_4 & 1\mathcal{A}_{\mathbb{F}_p} \\
\hline
 & & 17\mathcal{A}_{\mathbb{F}_p} + 8\mathcal{S}_{\mathbb{F}_p} + 2\mathcal{M}_{\mathbb{F}_p} + 0\mathcal{I}_{\mathbb{F}_p}
\end{array}
$$

Notes:

- 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.

Notes:

- Idea: our point arithmetic *ideally*
  1. **unified** $\Rightarrow \forall P, \textsc{Add}(P,P) = \textsc{Dbl}(P) = P + P = [2]P$
  2. **complete** $\Rightarrow \forall P, Q, \textsc{Add}(P,Q) = P + Q, \forall P, \textsc{Dbl}(P) = P + P = [2]P$

  such that

$$\begin{array}{lcr} \textsc{Dbl}(\mathcal{O}) & = & \mathcal{O} \\ \textsc{Add}(\mathcal{O},\mathcal{O}) & = & \mathcal{O} \\ \textsc{Add}(P,\mathcal{O}) & = & P \\ \textsc{Add}(\mathcal{O},Q) & = & Q \\ \textsc{Add}(P,P) & = & [2]P \end{array}$$

  but those we've looked at (clearly) *aren't* ...
- ... an implementation needs a set of special-cases to deal with each of the above.

Notes:

Algorithm (*complete* projective addition [3, Algorithm 1])

$\lambda_0 \leftarrow P_x \cdot Q_x$	$1\mathcal{M}_{\mathbb{F}_p}$	$R_z \leftarrow R_x + R_z$	$1\mathcal{A}_{\mathbb{F}_p}$
$\lambda_1 \leftarrow P_y \cdot Q_y$	$1\mathcal{M}_{\mathbb{F}_p}$	$R_x \leftarrow \lambda_1 - R_z$	$1\mathcal{A}_{\mathbb{F}_p}$
$\lambda_2 \leftarrow P_z \cdot Q_z$	$1\mathcal{M}_{\mathbb{F}_p}$	$R_z \leftarrow \lambda_1 + R_z$	$1\mathcal{A}_{\mathbb{F}_p}$
$\lambda_3 \leftarrow P_x + P_y$	$1\mathcal{A}_{\mathbb{F}_p}$	$R_y \leftarrow R_x \cdot R_z$	$1\mathcal{M}_{\mathbb{F}_p}$
$\lambda_4 \leftarrow Q_x + Q_y$	$1\mathcal{A}_{\mathbb{F}_p}$	$\lambda_1 \leftarrow \lambda_0 + \lambda_0$	$1\mathcal{A}_{\mathbb{F}_p}$
$\lambda_3 \leftarrow \lambda_3 \cdot \lambda_4$	$1\mathcal{M}_{\mathbb{F}_p}$	$\lambda_1 \leftarrow \lambda_1 + \lambda_0$	$1\mathcal{A}_{\mathbb{F}_p}$
$\lambda_4 \leftarrow \lambda_0 + \lambda_1$	$1\mathcal{A}_{\mathbb{F}_p}$	$\lambda_2 \leftarrow a_4 \cdot \lambda_2$	$1\mathcal{M}_{\mathbb{F}_p}$
$\lambda_3 \leftarrow \lambda_3 - \lambda_4$	$1\mathcal{A}_{\mathbb{F}_p}$	$\lambda_4 \leftarrow 3 \cdot a_6 \cdot \lambda_4$	$1\mathcal{M}_{\mathbb{F}_p}$
$\lambda_4 \leftarrow P_x + P_z$	$1\mathcal{A}_{\mathbb{F}_p}$	$\lambda_1 \leftarrow \lambda_1 + \lambda_2$	$1\mathcal{A}_{\mathbb{F}_p}$
$\lambda_5 \leftarrow Q_x + Q_z$	$1\mathcal{A}_{\mathbb{F}_p}$	$\lambda_2 \leftarrow \lambda_0 - \lambda_2$	$1\mathcal{A}_{\mathbb{F}_p}$
$\lambda_4 \leftarrow \lambda_4 \cdot \lambda_5$	$1\mathcal{M}_{\mathbb{F}_p}$	$\lambda_2 \leftarrow a_4 \cdot \lambda_2$	$1\mathcal{M}_{\mathbb{F}_p}$
$\lambda_5 \leftarrow \lambda_0 + \lambda_2$	$1\mathcal{A}_{\mathbb{F}_p}$	$\lambda_4 \leftarrow \lambda_4 + \lambda_2$	$1\mathcal{A}_{\mathbb{F}_p}$
$\lambda_4 \leftarrow \lambda_4 - \lambda_5$	$1\mathcal{A}_{\mathbb{F}_p}$	$\lambda_0 \leftarrow \lambda_1 \cdot \lambda_4$	$1\mathcal{M}_{\mathbb{F}_p}$
$\lambda_5 \leftarrow P_y + P_z$	$1\mathcal{A}_{\mathbb{F}_p}$	$R_y \leftarrow R_y + \lambda_0$	$1\mathcal{A}_{\mathbb{F}_p}$
$R_x \leftarrow Q_y + Q_z$	$1\mathcal{A}_{\mathbb{F}_p}$	$R_x \leftarrow \lambda_3 \cdot R_x$	$1\mathcal{M}_{\mathbb{F}_p}$
$\lambda_5 \leftarrow \lambda_5 \cdot R_x$	$1\mathcal{M}_{\mathbb{F}_p}$	$\lambda_0 \leftarrow \lambda_5 \cdot \lambda_4$	$1\mathcal{M}_{\mathbb{F}_p}$
$R_x \leftarrow \lambda_1 + \lambda_2$	$1\mathcal{A}_{\mathbb{F}_p}$	$R_x \leftarrow R_x - \lambda_0$	$1\mathcal{A}_{\mathbb{F}_p}$
$\lambda_5 \leftarrow \lambda_5 - R_x$	$1\mathcal{A}_{\mathbb{F}_p}$	$\lambda_0 \leftarrow \lambda_3 \cdot \lambda_1$	$1\mathcal{M}_{\mathbb{F}_p}$
$R_z \leftarrow a_4 \cdot \lambda_4$	$1\mathcal{M}_{\mathbb{F}_p}$	$R_z \leftarrow \lambda_5 \cdot R_z$	$1\mathcal{M}_{\mathbb{F}_p}$
$R_x \leftarrow 3 \cdot a_6 \cdot \lambda_2$	$1\mathcal{M}_{\mathbb{F}_p}$	$R_z \leftarrow R_z + \lambda_0$	$1\mathcal{A}_{\mathbb{F}_p}$

$$23\mathcal{A}_{\mathbb{F}_p} + 0\mathcal{S}_{\mathbb{F}_p} + 17\mathcal{M}_{\mathbb{F}_p} + 0\mathcal{I}_{\mathbb{F}_p}$$

Notes:

- **Idea**: given

$$E : y^2 = x^3 + a_4 x \quad + a_6$$

  one might select a_i to *optimise*, for example,
  - the curve (and hence group) order, or
  - point arithmetic.

Notes:

- 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.

Notes:

Part 2: in practice (14)
Curve parameters

- 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 $2\mathcal{S}_{\mathbb{F}_p}$.

© Daniel Page (csdsp@bristol.ac.uk)
Applied Cryptology
University of BRISTOL
git # c8178615 @ 2024-04-24

Notes:

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 $x \in \mathbb{G}^+$, a base-2 integer $0 \le y < n$
Output: The group element $r = [y]\,x \in \mathbb{G}^+$

```
r ← 0
for i = |y| − 1 downto 0 step −1 do
    r ← [2] r
    if y_i = 1 then
        r ← r + x
    end
end
return r
```

Notes:

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

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

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

Notes:

- **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

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

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

Definition

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

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

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 consequtive digits in $\hat{y}$ are non-zero, i.e., if $\hat{y}_i \neq 0$ then either $\hat{y}_{i+1} = 0$ and/or $\hat{y}_{i-1} = 0$.

Notes:

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

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

  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

```
y' ← ∅, i ← 0
while y ≥ 1 do
    if y ≡ 1 (mod 2) then
        y'_i ← 2 − (y mod 4), y ← y − y'_i
    else
        y'_i ← 0
    end
    y ← ⌊y/2⌋, i ← i + 1
end
return y'
```

Algorithm (1Mul-L2R-NAF)

Input: A group element $x \in \mathbb{G}^+$, a base-2 integer $0 \le y < n$
Output: The group element $r = [y]\,x \in \mathbb{G}^+$

```
y' ← Recode-NAF(y)
r ← 0
for i = |y'| − 1 downto 0 step −1 do
    r ← [2] r
    if y'_i = +1 then
        r ← r + x
    else if y'_i = −1 then
        r ← r − x
    end
end
return r
```

© Daniel Page (csdsp@bristol.ac.uk)
Applied Cryptology
University of BRISTOL
git # c8178615 @ 2024-04-24

Notes:

Conclusions

Notes:

- Take away points: you can often simple *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

Notes:

[1] D.J. Bernstein and T. Lange. *ECCHacks*. URL: http://ecchacks.cr.yp.to (see pp. 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35).

[2] D.J. Bernstein and T. Lange. *Explicit-Formulas Database (EFD)*. URL: http://www.hyperelliptic.org/EFD (see p. 103).

[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 (see p. 109).

[4] J.A. Solinas. *Generalized Mersenne Numbers*. Tech. rep. CORR 99-39. Centre for Applied Cryptographic Research (CACR), University of Waterloo, 1999 (see p. 85).

[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. 75, 77, 79, 81).

[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. 83).