Chapter 2

Elliptic curves as cryptographic groups

The purpose of this chapter is to introduce elliptic curves as they are used in cryptography. Put simply, an elliptic curve is an abstract type of *group*.

Perhaps a newcomer will find this abstractness apparent immediately when we insist that to understand elliptic curve groups in cryptography, the reader should be familiar with the basics of *finite fields* $\mathbb{F}_q$. This is because, more generally, elliptic curves are groups which are defined on top of (over) fields. Even though elliptic curve groups permit only one binary operation (the so called *group law*), the operation itself is computed within the *underlying field*, which by definition permits two operations (and their inverses). For a general field K, the group elements of an elliptic curve E are *points* whose (x, y) coordinates come from $\overline{K}$ (the algebraic closure of K), and which satisfy the (affine) curve equation for E, given as

$$E : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6, \tag{2.1}$$

where $a_1, ..., a_6 \in \overline{K}$. Equation (2.1) is called the *general Weierstrass equation* for elliptic curves. Aside from all the $(x, y) \in \overline{K}$ solutions to the equation above, there is one extra point which can not be defined using the affine equation, but which must be included to complete the group definition. This point is called the *point at infinity*, which we denote by $\mathcal{O}$, and we will define it properly in a

moment.

If $a_1, ..., a_6 \in K$, then we say E is *defined over* K, and write this as E/K (the same goes for any extension field L of K). Before we go any further, we make a convenient simplification of the general Weierstrass equation. If the field characteristic is not 2 or 3, then divisions by 2 and 3 in K permit the substitutions $y \mapsto (y - a_1x - a_3)/2$ to give $E : y^2 = 4x^3 + b_2x^2 + 2b_4x + b_6$, and then $(x, y) \mapsto \left(\frac{x-3b_2}{36}, \frac{y}{108}\right)$, which (upon appropriate rescaling) yields the following simplified equation.

$$E : y^2 = x^3 + ax + b. \tag{2.2}$$

Equation (2.2) is called the *short Weierstrass equation* for elliptic curves, and will be used all the way through this text. Namely, we will always be working over large prime fields, where the short Weierstrass equation covers all possible isomorphism classes of elliptic curves, so the curves we use will always be an instance of (2.2).

Example 2.0.1 (Magma script). $E/\mathbb{Q} : y^2 = x^3 - 2$ is an elliptic curve. Along with the point at infinity $\mathcal{O}$ (which we are still yet to define), the set of points over $\mathbb{Q}$ is written as $E(\mathbb{Q})$, and is defined as $E(\mathbb{Q}) = \{(x, y) \in \mathbb{A}^2(\mathbb{Q}) : y^2 = x^3 - 2\} \cup \{\mathcal{O}\}$. The point $P = (x_P, y_P) = (3, 5)$ lies in $E(\mathbb{Q})$, as do $Q = (x_Q, y_Q) = \left(\frac{129}{100}, \frac{-383}{1000}\right)$ and $R = (x_R, y_R) = \left(\frac{164323}{29241}, \frac{-66234835}{5000211}\right)$, so we can write $P, Q, R \in E(\mathbb{Q})$. We usually write E to represent the group of points over the full algebraic closure, so for example, the point $S = (x_S, y_S) = \left(0, \sqrt{-2}\right) \in E = E(\overline{\mathbb{Q}})$, but $S \notin E(\mathbb{Q})$. Soon we will be defining the binary group operation $\oplus$ on E using rational formulas in the underlying field, so an active reader can return to this example with these formulas to verify that $R = P \oplus Q$, where x_R, y_R are computed from x_P, y_P, x_Q, y_Q using additions and multiplications (also subtractions and inversions) in $\mathbb{Q}$. Furthermore, it can also be verified that $Q = P \oplus P$, so that $R = P \oplus P \oplus P$; we usually write these as $Q = [2]P$ and $R = [3]P$, where $\underbrace{P \oplus P \cdots \oplus P}_{n} = [n]P$ in general. To finish this example, we remark that if $(x', y') \in E$, then $(x', -y') \in E$ (but is not distinct if $y' = 0$), which is true for any elliptic curve in short Weierstrass form.

Example 2.0.2 (Magma script). $E/\mathbb{F}_{11} : y^2 = x^3 + 4x + 3$ is an elliptic curve. $E(\mathbb{F}_{11})$ has 14 points: $(0, 5)$, $(0, 6)$, $(3, 3)$, $(3, 8)$, $(5, 4)$,$(5, 7)$, $(6, 1)$, $(6, 10)$, $(7, 0)$, $(9, 3)$, $(9, 8)$, $(10, 3)$, $(10, 8)$, not forgetting the point at infinity $\mathcal{O}$. Notice that all

but two points come in pairs (x', y') and $(x', -y')$, the exceptions being $(x', y') = (7, 0)$ (since $y' = -y' = 0$) and $\mathcal{O}$. If we form the quadratic extension $\mathbb{F}_{q^2} = \mathbb{F}_q(i)$ with $i^2 + 1 = 0$, then considering E over $\mathbb{F}_{q^2}$ will allow many more solutions, and give many more points: namely, $\#E(\mathbb{F}_{q^2}) = 140$. In addition to the points in $E(\mathbb{F}_q)$, $E(\mathbb{F}_{q^2})$ will also contain those points with x-coordinates in $\mathbb{F}_q$ that did not give x^3+4x+3 as a quadratic residue in $\mathbb{F}_q$ (but necessarily do in $\mathbb{F}_{q^2}$), and many more with both coordinates in $\mathbb{F}_{q^2} \setminus \mathbb{F}_q$. Examples of both such points are $(2, 5i)$ and $(2i + 10, 7i + 2)$ respectively. It is not a coincidence that $\#E(\mathbb{F}_q) \mid \#E(\mathbb{F}_{q^2})$, since $E(\mathbb{F}_q)$ is a subgroup of $E(\mathbb{F}_{q^2})$.

Not every tuple $(a, b) \in K \times K$ gives rise to the curve given by $f(x, y) = y^2 - (x^3 + ax + b) = 0$ being an elliptic curve. If there exists $P = (x_P, y_P)$ on f such that both partial derivatives $\frac{\partial f}{\partial x}$ and $\frac{\partial f}{\partial y}$ vanish simultaneously at P, then P is called a *singular* point and f is also deemed singular. Conversely, if no such point exists, f is called *non-singular*, or *smooth*, and is then an elliptic curve. It is easy enough to show that a singularity occurs if and only if $4a^3 + 27b^2 = 0$ (see [Sil09, Ch. III.1, Prop. 1.4]), so as long as $4a^3 + 27b^2 \neq 0$ in K, then $E/K : y^2 = x^3 + ax + b$ is an elliptic curve.

In cryptography we only ever instantiate elliptic curves defined over finite fields, but it is often conceptually helpful to view graphs of elliptic curves over $\mathbb{R}$. We illustrate the difference between singular and non-singular (smooth) elliptic curves in Figures 2.1-2.4.

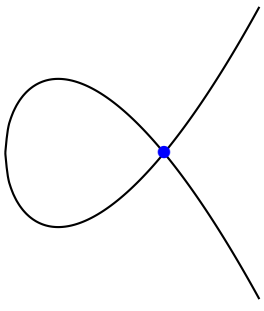


Figure 2.1: Singular curve $y^2 = x^3 - 3x + 2$ over $\mathbb{R}$.

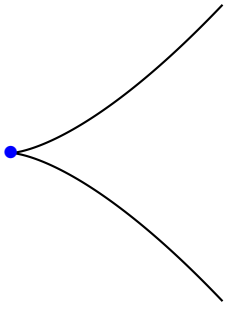


Figure 2.2: Singular curve $y^2 = x^3$ over $\mathbb{R}$.

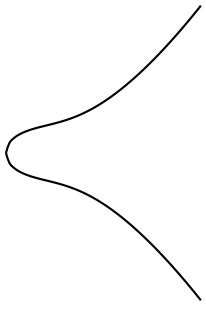


Figure 2.3: Smooth curve $y^2 = x^3 + x + 1$ over $\mathbb{R}$.

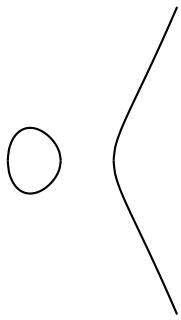


Figure 2.4: Smooth curve $y^2 = x^3 - x$ over $\mathbb{R}$.

2.1 The group law: the chord-and-tangent rule

We now turn to describing the elliptic curve group law, and it is here that viewing pictures of elliptic curves over $\mathbb{R}$ is especially instructive. We start with a less formal description until we define the role of the point at infinity $\mathcal{O}$. The group law exploits the fact that, over any field, a line (a degree one equation in x and y) intersects a cubic curve (a degree three equation in x and y) in three places (this is a special case of a more general theorem due to Bezout [Har77, I.7.8]). Namely, if we run a line $\ell : y = \lambda x + \nu$ between two points $P = (x_P, y_P)$ and $Q = (x_Q, y_Q)$ on E, then substituting this line into $E : y^2 = x^3 + ax + b$ will give a cubic polynomial in x, the roots of which are the x-coordinates of the three points of intersection between ℓ and E. Knowing the two roots (x_P and x_Q) allows us to determine a unique third root that corresponds to the third and only other point in the affine intersection $\ell \cap E$, which we denote by $\ominus R$ (the reason will become clear in a moment). The point $\ominus R$ is then "flipped" over the x-axis to the point R. In general, the elliptic curve composition law $\oplus$ is defined by this process, namely $R = P \oplus Q$. When computing $R = P \oplus P$, the line ℓ is computed as the tangent to E at P. That is, the derivatives of ℓ and E are matched at P, so (counting multiplicities) ℓ intersects E "twice" at P. Figures 2.5 and 2.6 illustrate why this process is aptly named the *chord-and-tangent* rule.

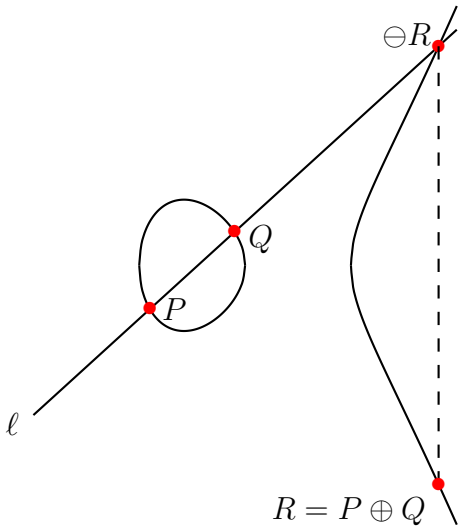


Figure 2.5: Elliptic curve addition.

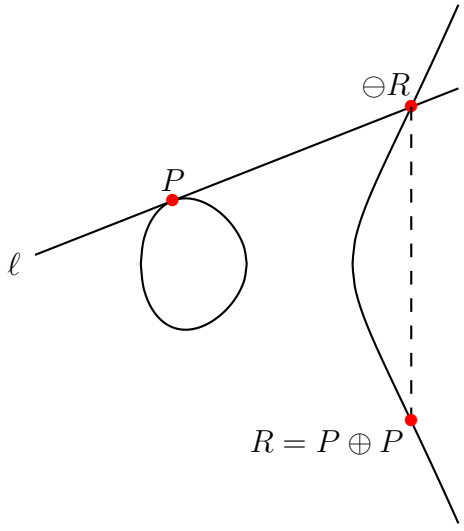


Figure 2.6: Elliptic curve doubling.

Having loosely defined the general group operation, we can now (also loosely)

define the role of the point at infinity $\mathcal{O}$. To try and place it somewhere in the above diagrams, one can think of $\mathcal{O}$ as being a point that simultaneously sits infinitely high and infinitely low in the y direction. This allows us to informally conceptualise two properties of elliptic curve groups: firstly, that the point at infinity $\mathcal{O}$ plays the role of the *identity* of the group; and secondly, that the unique inverse of a point is its reflected image over the x-axis (e.g. the $\ominus R$'s in Figures 2.5 and 2.6 are the respective inverses of the R's, and vice versa). If we apply the process in the previous paragraph to compute $R \oplus (\ominus R)$, we start by finding the vertical line that connects them (the dashed lines in Figures 2.5 and 2.6). This line also intersects E (twice) at the point at infinity $\mathcal{O}$, which is then reflected back onto itself, giving $R \oplus (\ominus R) = \mathcal{O}$. Thus, if we define the identity of the group to be $\mathcal{O}$, then the inverse of any element $R = (x_R, y_R)$ is taken as $\ominus R = (x_R, -y_R)$.

Example 2.1.1 (Magma script). $E/\mathbb{R} : y^2 = x^3 - 2x$ is an elliptic curve. The points $(-1,-1)$, $(0,0)$ and $(2,2)$ are all on E, and are also on the line $\ell : y = x$. Applying the technique described above to compute some example group law operations via the line ℓ, we have $(-1,-1) \oplus (0,0) = (2,-2)$, $(2,2) \oplus (0,0) = (-1,1)$, and $(-1,-1) \oplus (2,2) = (0,0)$. All but four points come in pairs with their inverse (i.e. (x', y') and $(x', -y')$); the exceptions being $(0,0)$, $(\sqrt{2}, 0)$, $(-\sqrt{2}, 0)$ (notice the vertical tangents when $y = 0$ in these cases), and $\mathcal{O}$, which are all their own inverse, e.g. $(0,0) = \ominus(0,0)$, so $(0,0) \oplus (0,0) = \mathcal{O}$ on E. The tangent line ℓ' to E at $(-1,-1)$ is $\ell' : y = -\frac{1}{2}x - \frac{3}{2}$, and it intersects E once more at $(\frac{9}{4}, -\frac{21}{8})$, which gives $(-1,-1) \oplus (-1,-1) = [2](-1,-1) = (\frac{9}{4}, \frac{21}{8})$.

Example 2.1.2 (Magma script). In this example we consider the same curve equation as the last example, but this time over a small finite field, namely $E/\mathbb{F}_{11} : y^2 = x^3 - 2x$. Rational points are injected naturally across to the finite field case (as long as there is no conflict with the characteristic), so we can immediately find the points $(0,0)$, $(2,2)$ and $(-1,-1) = (10,10)$ (and their inverses) in Figure 2.9. In this case, consider performing the group law operation between the (blue) points $(5,7)$ and $(8,10)$. The line ℓ that joins them is $y = x + 2$, which intersects E once more at $(10,1)$. Negating the y-coordinate finds the other point on the dashed line, and gives $(5,7) \oplus (8,10) = (10,10)$.

Example 2.1.2 is also intended to justify why, although (in cryptography) we only ever use elliptic curves over finite fields, we often opt to illustrate the group law by drawing the continuous pictures of curves over $\mathbb{R}$.

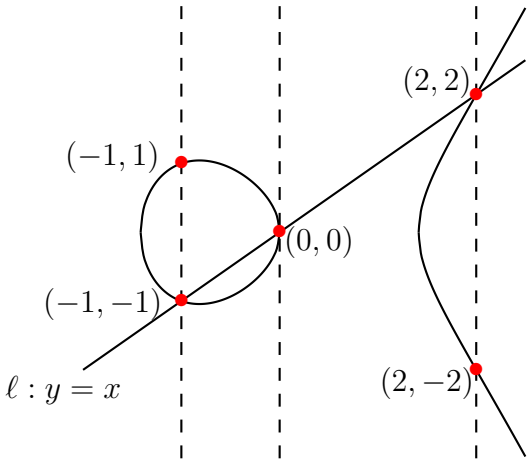


Figure 2.7: Addition in $\mathbb{R}$.

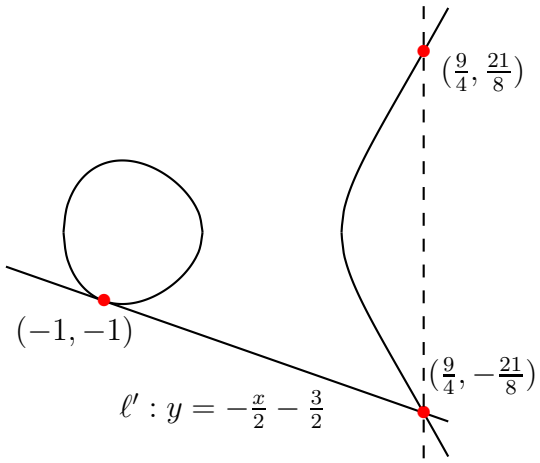


Figure 2.8: Doubling in $\mathbb{R}$.

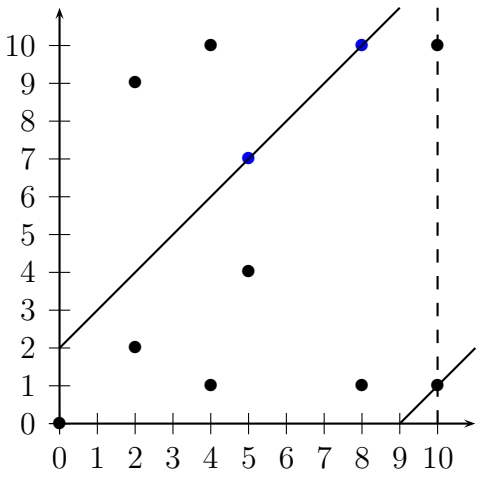
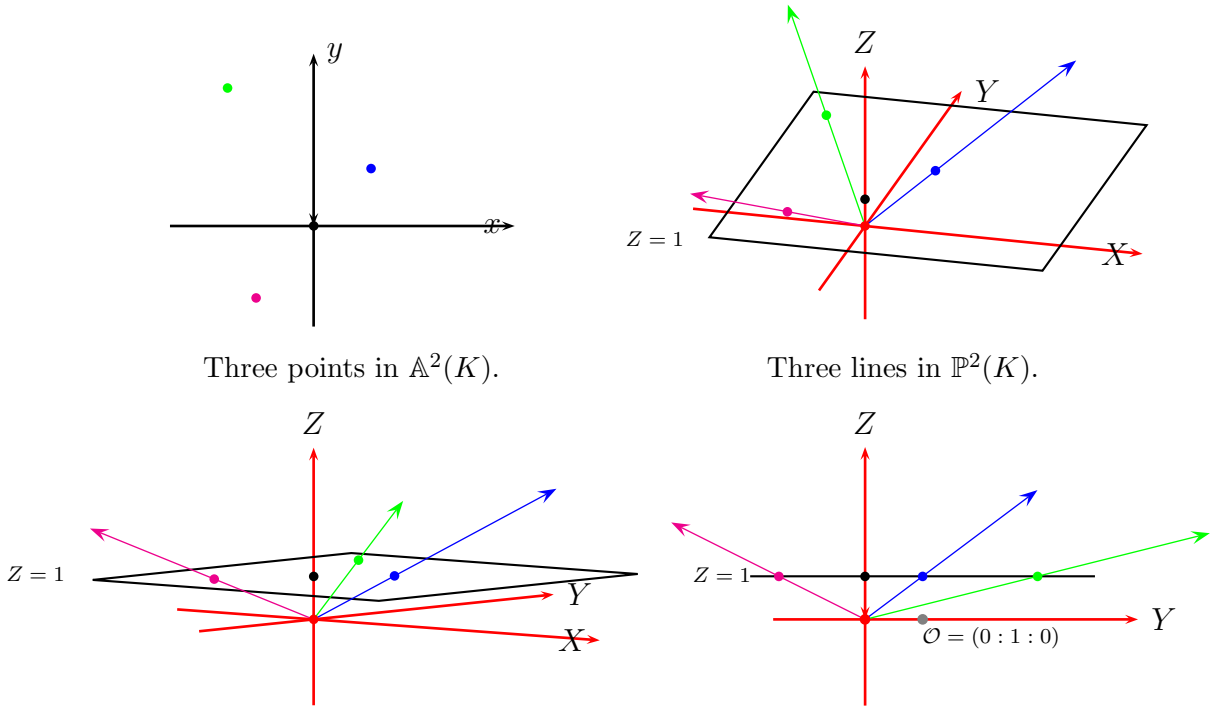


Figure 2.9: The points (excluding $\mathcal{O}$) on $E(\mathbb{F}_{11})$.

2.1.1 The point at infinity in projective space

We now focus our attention on giving a more formal definition for the point at infinity. So far we have been describing elliptic curves in *affine space* as a set of affine points together with the point at infinity: $E = \{(x, y) \in \mathbb{A}^2(\overline{K}) : y^2 = x^3 + ax + b\} \cup \{\mathcal{O}\}$. In general, a more precise way to unify (or include) points at infinity with the affine points is to work in *projective space*: essentially, instead of working with points in n-space, we work with lines that pass through the origin in $(n+1)$-space. For our purposes, this means our affine points in 2-space become lines in 3-space, namely that $(x, y) \in \mathbb{A}^2(\overline{K})$ corresponds to the line defined by all points of the form $(\lambda x, \lambda y, \lambda) \in \mathbb{P}^2(\overline{K})$, where $\lambda \in \overline{K}^*$. That is, $\mathbb{P}^2$ is $\mathbb{A}^3 \setminus$

$\{(0,0,0)\}$ modulo the following congruence condition: $(x_1, y_1, z_1) \sim (x_2, y_2, z_2)$ if there exists $\lambda \in \overline{K}^*$ such that $(x_1, y_1, z_1) = (\lambda x_2, \lambda y_2, \lambda z_2)$. Figure 2.10 illustrates the relationship between points in $\mathbb{A}^2$ with their congruence classes (lines) in $\mathbb{P}^2$; the lines in 3-space should also extend "downwards" into the region where $Z < 0$ but we omitted this to give more simple pictures. We reiterate that these lines do not include the point $(0, 0, 0)$.



Three points in $\mathbb{A}^2(K)$. Three lines in $\mathbb{P}^2(K)$.

Three lines in $\mathbb{P}^2(K)$. Three lines in $\mathbb{P}^2(K)$.

Figure 2.10: Identifying points in $\mathbb{A}^2$ with lines in $\mathbb{P}^2$

We usually use capital letters and colons to denote a (representative of a) congruence class in projective coordinates, so that in general $(X : Y : Z)$ represents the set of all points on the "line" in $\mathbb{P}^2$ that correspond to $(x, y) \in \mathbb{A}^2$. There are many copies of $\mathbb{A}^2$ in $\mathbb{P}^2$, but we traditionally map the affine point $(x, y) \in \mathbb{A}^2$ to projective space via the trivial inclusion $(x, y) \mapsto (x : y : 1)$, and for any $(X : Y : Z) \neq \mathcal{O} \in \mathbb{P}^2$, we map back to $\mathbb{A}^2$ via $(X : Y : Z) \mapsto (X/Z, Y/Z)$. The point at infinity $\mathcal{O}$ is represented by $(0 : 1 : 0)$ in projective space (see the last diagram in Figure 2.10), for which we immediately note that the map back to $\mathbb{A}^2$ is ill-defined.

Example 2.1.3 (Magma script). $E/\mathbb{R} : y^2 = x^3 + 3x$ is an elliptic curve. $P =$

$(3,6) \in \mathbb{A}^2(\overline{\mathbb{R}})$ is a point on E. In projective space, P becomes $P = (3 : 6 : 1) \in \mathbb{P}^2(\overline{\mathbb{R}})$, which represents all points in $(3\lambda, 6\lambda, \lambda)$ for $\lambda \in \overline{\mathbb{R}} \setminus \{0\}$. For example, the points $(12, 24, 4)$, $(-3\sqrt{-1}, -6\sqrt{-1}, -1\sqrt{-1})$, $(3\sqrt{2}, 6\sqrt{2}, \sqrt{2})$ in $\mathbb{A}^3(\overline{\mathbb{R}})$ are all equivalent (modulo the congruence condition) in $\mathbb{P}^2(\overline{\mathbb{R}})$, where they are represented by P. As usual, the point at infinity on E is $\mathcal{O} = (0 : 1 : 0)$.

The way we define the collection of points in projective space is to *homogenise* $E : y^2 = x^3 + ax + b$ by making the substitution $x = X/Z$ and $y = Y/Z$, and multiplying by Z^3 to clear the denominators, which gives

$$E_{\mathbb{P}} : Y^2Z = X^3 + aXZ^2 + bZ^3. \tag{2.3}$$

The set of points (X, Y, Z) with coordinates in $\overline{K}$ that satisfies (2.3) is called the *projective closure* of E. Notice that $(0, \lambda, 0)$ is in the projective closure for all $\lambda \in \overline{K}^*$, and that all such points cannot be mapped into $\mathbb{A}^2$, justifying the representative of point at infinity being $\mathcal{O} = (0 : 1 : 0)$.

Example 2.1.4 (Magma script). Consider $E/\mathbb{F}_{13} : y^2 = x^3 + 5$. There are 15 affine points $(x, y) \in \mathbb{A}^2(\mathbb{F}_{13})$ on E, which (with the point at infinity $\mathcal{O}$) gives $\#E(\mathbb{F}_{13}) = 16$. On the other hand, if we homogenise (or projectify) E to give $E_{\mathbb{P}}/\mathbb{F}_{13} : Y^2Z = X^3 + 5Z^3$, then there are 16 classes $(X : Y : Z) \in \mathbb{P}^2(\mathbb{F}_{13})$: $(0 : 1 : 0)$, $(2 : 0 : 1)$, $(4 : 2 : 1)$, $(4 : 11 : 1)$, $(5 : 0 : 1)$, $(6 : 0 : 1)$, $(7 : 6 : 1)$, $(7 : 7 : 1)$, $(8 : 6 : 1)$, $(8 : 7 : 1)$, $(10 : 2 : 1)$, $(10 : 11 : 1)$, $(11 : 6 : 1)$, $(11 : 7 : 1)$, $(12 : 2 : 1)$, $(12 : 11 : 1)$. Each of these classes represents several points $(X, Y, Z) \in \mathbb{A}^3(\mathbb{F}_{13})$ whose coordinates satisfy $Y^2Z = X^3 + 5Z^3$ (there are actually 195 such points, but this is not important). In fact, each class represents infinitely many points on $E_{\mathbb{P}}(\overline{\mathbb{F}}_{13})$. Any reader that is familiar with Magma, or has been working through our examples with the accompanying Magma scripts, will recognise the representation of points as representatives in $\mathbb{P}^2$.

The projective coordinates (X, Y, Z) used to replace the affine coordinates (x, y) above are called *homogenous projective coordinates*, because the projective version of the curve equation in (2.3) is homogeneous. These substitutions ($x = X/Z$, $y = Y/Z$) are the most simple (and standard) way to obtain projective coordinates, but we are not restricted to this choice of substitution. For example, many papers in ECC have explored more general substitutions of the form $x = X/Z^i$ and $y = Y/Z^j$ on various elliptic curves [BL07a].

Example 2.1.5 (Magma script). Consider $E/\mathbb{F}_{41} : y^2 = x^3 + 4x - 1$. Using

homogeneous coordinates gives rise to the projective equation $Y^2Z = X^3 + 4XZ^2 - Z^3$, with the point at infinity being $\mathcal{O} = (0 : 1 : 0)$. An alternative projection we can use is $x = X/Z$ and $y = Y/Z^2$, which in this instance give the projective equation $Y^2 = X^3Z + 4XZ^3 - Z^4$, from which the point at infinity is seen (from putting $Z = 0$) to be $\mathcal{O} = (1 : 0 : 0)$. Another commonly used coordinate system is Jacobian coordinates, which use the substitutions $x = X/Z^2$ and $y = Y/Z^3$ to give the projective equation $Y^2 = X^3 + 4XZ^4 - Z^6$. In this case, we substitute $Z = 0$ to see that the point at infinity is defined by the line $\mathcal{O} = (\lambda^2 : \lambda^3 : 0) \in \mathbb{P}^2(\mathbb{F}_{41})$.

2.1.2 Deriving explicit formulas for group law computations

We are now in a position to give explicit formulas for computing the elliptic curve group law. The chord-and-tangent process that is summarised in Figures 2.5 and 2.6 allows a simple derivation of these formulas. We derive the formulas in affine space, but will soon transfer them into projective space as well. The derivation of the formulas for point additions $R = P \oplus Q$ and for point doublings $R = P \oplus P$ follow the same recipe, the main difference being in the calculation of the gradient λ of the line $\ell : y = \lambda x + \nu$ that is used. We will first derive the formulas for the addition $R = P \oplus Q$ in the general case, and will then make appropriate changes for the general doubling formulas. By "general case", we mean group law operations between points where neither point is $\mathcal{O}$, and the points that are being added are not each inverses of one another; we will handle these special cases immediately after the general cases. Referring back to Figure 2.5, the line $\ell : y = \lambda x + \nu$ that intersects $P = (x_P, y_P)$ and $Q = (x_Q, y_Q)$ has gradient $\lambda = (y_Q - y_P)/(x_Q - x_P)$. From here, ν can simply be calculated as either $\nu = y_P - \lambda x_P$ or $\nu = y_Q - \lambda x_Q$, but in the literature we will often see an unbiased average of the two as $\nu = (y_Q x_P - y_P x_Q)/(x_P - x_Q)$. From here we substitute $\ell : y = \lambda x + \nu$ into $E : y^2 = x^3 + ax + b$ to find the third affine point of intersection, $\ominus R$, in $\ell \cap E$. Finding the coordinates of $\ominus R$ trivially reveals the coordinates of $R = (x_R, y_R)$, since $\ominus R = (x_R, -y_R)$; the roots of the cubic that

result will be x_P, x_Q and x_R. Namely,

$$\begin{aligned}(x - x_P)(x - x_Q)(x - x_R) &= (x^3 + ax + b) - (\lambda x + \nu)^2 \\ &= x^3 - \lambda^2 x^2 + (a - 2\lambda\nu)x + b - \nu^2.\end{aligned}$$

We only need to look at the coefficient of x^2 to determine x_R, since the coefficient on the left hand side is $-(x_P + x_Q + x_R)$. From here, recovering the y-coordinate is simple, since $-y_R$ lies on ℓ, so

$$x_R = \lambda^2 - x_P - x_Q; \qquad\qquad y_R = -(\lambda x_R + \nu).$$

This finishes the description of addition in the general case. When adding P to itself (i.e. doubling P – refer back to Figure 2.6), the line $\ell : y = \lambda x + \nu$ is the tangent to E at P. Thus, its gradient λ is the derivative function dy/dx of E, evaluated at P. To obtain dy/dx, we differentiate the curve equation implicitly, as

$$\begin{aligned}\frac{d}{dx}(y^2) &= \frac{d}{dx}(x^3 + ax + b) \\ \frac{d}{dy}(y^2)\frac{dy}{dx} &= 3x^2 + a \\ \frac{dy}{dx} &= \frac{3x^2 + a}{2y}.\end{aligned}$$

Thus, $\lambda = \frac{dy}{dx}(P) = (3x_P^2 + a)/(2y_P)$, and $\nu = y_P - \lambda x_P$. Again, we substitute ℓ into E, but this time two of the roots of the resulting cubic are x_P, so we obtain x_R and y_R as

$$x_R = \lambda^2 - 2x_P; \qquad\qquad y_R = -(\lambda x_R + \nu).$$

This finishes the derivation of doubling formulas in the general case. We now complete the group law description by looking at the special cases. The point at infinity $\mathcal{O}$ is the identity, or neutral element, so any operation involving it is trivial. Otherwise, any operation between elements P and Q with different x-coordinates employs the general addition. This leaves the remaining cases of $x_P = x_Q$: (i) if $y_P = -y_Q$, then P and Q are inverses of each other and $P \oplus Q = \mathcal{O}$ (note that this includes $y_P = y_Q = 0$), and (ii) if $y_P = y_Q \neq 0$, then $P = Q$ and we use the point doubling formulas.

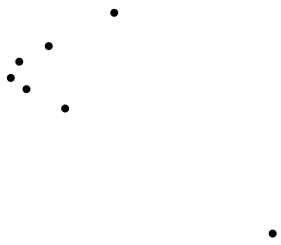
Much of the literature concerning the elliptic curve group law tends to present the complete description in the previous paragraph using an "if-then-else" style algorithm, where the "if" statements distinguish which of the above scenarios we are in. In optimised cryptographic implementations however, this is not the way that the group law operation is coded. This is because the groups we use are so large that the chances of running into a special case (that is not general doubling or general addition) randomly is negligible. Moreover, the parameters are usually chosen so that we are guaranteed not to run into these cases. In this light then, it will soon become clear that the major operations we are concerned with are point additions $R = P \oplus Q$ and point doublings $R = P \oplus P$, the formulas for which are summarised in (2.4) and (2.5) respectively.

$$\text{(Affine addition)} \qquad \lambda = \frac{y_Q - y_P}{x_Q - x_P}; \qquad \nu = y_P - \lambda x_P;$$
$$(x_P, y_P) \oplus (x_Q, y_Q) = (x_R, y_R) = \left(\lambda^2 - x_P - x_Q, -(\lambda x_R + \nu)\right). \quad (2.4)$$

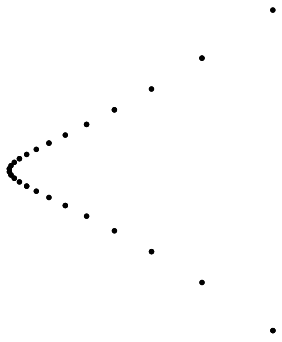
$$\text{(Affine doubling)} \qquad \lambda = \frac{3x_P^2 + a}{2y_P}; \qquad \nu = y_P - \lambda x_P;$$
$$[2](x_P, y_P) = (x_P, y_P) \oplus (x_P, y_P) = (x_R, y_R) = \left(\lambda^2 - 2x_P, -(\lambda x_R + \nu)\right). \quad (2.5)$$

Example 2.1.6 (Magma script). We revisit the curve $E/\mathbb{Q} : y^2 = x^3 - 2$ from Example 2.0.1 to verify the group law calculations that were stated. We start with the point doubling of $P = (x_P, y_P) = (3, 5)$, to compute $Q = [2]P = P \oplus P$ using (2.5). Here, $\lambda = \frac{3x_P^2+a}{2y_P} = \frac{3\cdot 3^2+0}{2\cdot 5} = \frac{27}{10}$, from which ν follows as $\nu = y_P - \lambda x_P = 5 - \frac{27}{10} \cdot 3 = -\frac{31}{10}$. Thus, $x_Q = \lambda^2 - 2x_P = (\frac{27}{10})^2 - 2 \cdot 3 = \frac{129}{100}$, and $y_Q = -(\lambda x_Q + \nu) = -(\frac{27}{10} \cdot \frac{129}{100} - \frac{31}{10}) = -\frac{383}{1000}$, giving $(x_Q, y_Q) = [2](x_P, y_P) = (\frac{129}{100}, -\frac{383}{1000})$. For the addition $R = P \oplus Q$, we use the formulas in (2.4), so $\lambda = \frac{y_Q - y_P}{x_Q - x_P} = (-\frac{383}{1000} - 5)/(\frac{129}{100} - 3) = \frac{5383}{1710}$, and $\nu = y_P - \lambda x_P = 5 - \frac{5383}{1710} \cdot 3 = -\frac{2533}{570}$. Thus, $x_R = \lambda^2 - x_P - x_Q = (\frac{5383}{1710})^2 - 3 - \frac{129}{100} = \frac{164323}{29241}$, and $y_R = \lambda x_R + \nu = \frac{5383}{1710} \cdot \frac{164323}{29241} - \frac{2533}{570} = -\frac{66234835}{5000211}$, so $(x_R, y_R) = (\frac{164323}{29241}, -\frac{66234835}{5000211})$. Since $Q = [2]P = P \oplus P$, then $R = P \oplus Q = [3]P$. We finish this example with a remark that further justifies the use of finite fields as the underlying fields in cryptography. It is not too painful to show that $P = (3, 5)$ and $\ominus P = (3, -5)$ are the only integral points on E [Sil09, Ch. IX, Prop. 7.1(b)], or that $E(\mathbb{Q})$ is actually *infinite cyclic* [Sil09, Ch. IX, Remark 7.1.1], meaning that among

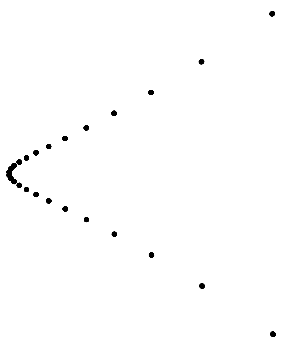
infinitely many rational points, only two have integer coordinates. Besides the infinite nature of $E(\mathbb{Q})$ (the lack of any finite subgroups is not useful in the context of discrete logarithm based cryptographic groups), observing the growing size of the numerators and denominators in $[n]P$, even for very small values of n, shows why using $E(\mathbb{Q})$ would be impractical. Using Magma, we can see that the denominator of the y-coordinate of $[10]P$ is 290 bits, whilst the denominator in $[100]P$ is 29201 bits, which agrees with the group law formulas in (2.4) and (2.5) that suggest that denominators of successive scalar multiples of P would grow quadratically; even Magma takes its time computing $[1000]P$, whose denominator is 2920540 bits, and Magma could not handle the computation of $[10000]P$. In Figure 2.11 we plot multiples of $P = (3, 5)$ that fall within the domain $x < 6$.



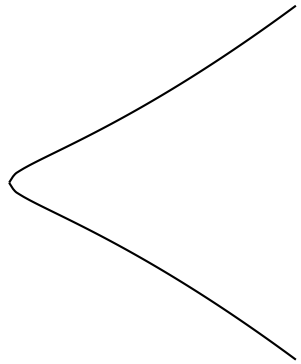
Of the first 10 multiples of $P = (3, 5)$ in $E(\mathbb{Q})$, 7 had $x < 6$.



Of the first 100 multiples of $P = (3, 5)$ in $E(\mathbb{Q})$, 64 had $x < 6$.



Of the first 1000 multiples of $P = (3, 5)$ in $E(\mathbb{Q})$, 635 had $x < 6$.



$E : y^2 = x^3 - 2$ over $\mathbb{R}$.

Figure 2.11: More and more points (with $x < 6$) in the infinite group $E(\mathbb{Q})$

From now on we will only be working with elliptic curves over finite fields. We start with a simple example of basic group law computations on $E(\mathbb{F}_q)$ to

summarise the discussion up until this point.

Example 2.1.7 (Magma script). $E/\mathbb{F}_{23} : y^2 = x^3 + 5x + 7$ is an elliptic curve, and both $P = (x_P, y_P) = (2, 5)$ and $Q = (x_Q, y_Q) = (12, 1)$ are on E. Using the affine point addition formulas in (2.4), we find $R = P \oplus Q$ by first computing $\lambda = \frac{y_Q - y_P}{x_Q - x_P} = \frac{1-5}{12-2} = -4 \cdot 10^{-1} = -28 = 18$, from which ν follows as $\nu = y_P - \lambda x_P = 5 - 18 \cdot 2 = -31 = 15$, so $\ell : y = 18x + 15$ is the line running through P and Q. We then compute $(x_R, y_R) = (\lambda^2 - x_P - x_Q, -(\lambda x_R + \nu))$, so $x_R = 18^2 - 2 - 12 = 11$ and $y_R = -(18 \cdot 11 + 15) = 17$, meaning $R = (11, 17)$. Applying (2.5) to compute $S = [2]P$ gives $\lambda' = \frac{3x_P^2 + a}{2y_P} = \frac{3 \cdot 2^2 + 5}{2 \cdot 5} = 17 \cdot 10^{-1} = 17 \cdot 7 = 4$, and ν' follows as $\nu' = y_P - \lambda' x_P = 5 - 4 \cdot 2 = 20$, so $\ell' : y = 4x + 20$ is the tangent line that intersects E with multiplicity two at P. We then compute $(x_S, y_S) = (\lambda'^2 - 2x_P, -(\lambda' x_S + \nu'))$, so $x_S = 4^2 - 2 \cdot 2 = 12$ and $y_S = -(4 \cdot 12 + 20) = -68 = 1$, meaning $S = (12, 1)$.

We now give an example of the *multiplication-by-m* map on E, defined as

$$[m] : E \to E, \qquad P \mapsto [m]P,$$

and illustrate the straightforward way to compute it in practice. This operation is analogous to exponentiation $g \mapsto g^m$ in $\mathbb{Z}_q^*$, and is the central operation in ECC, as it is the *one-way* operation that buries discrete logarithm problems in $E(\mathbb{F}_q)$. To efficiently compute the exponentiation g^m in $\mathbb{Z}_q^*$, we *square-and-multiply*, whilst to compute the scalar multiplication $[m]P$ in $E(\mathbb{F}_q)$, we (because of the additive notation) *double-and-add*.

Example 2.1.8 (Magma script). Let $E/\mathbb{F}_{1021} : y^2 = x^3 - 3x - 3$ so that $r = \#E(\mathbb{F}_q) = 1039$ is prime. Let $P = (379, 1011) \in E$ and $m = 655$, and suppose we are to compute $[m]P = [655](379, 1011)$. To double-and-add, we write the (10-bit) binary representation of m as $m = (m_9, ..., m_0)_2 = (1, 0, 1, 0, 0, 0, 1, 1, 1, 1)$. Initialising $T \leftarrow P$, and starting from the second most significant bit m_8, we successively compute $T \leftarrow [2]T$ for each bit down to m_0, and whenever $m_i = 1$ we compute $T \leftarrow T + P$. So, in our case it takes 9 doublings $T \leftarrow [2]T$ and 5 additions $T \leftarrow T + P$ to compute $[m]P$, which ends up being $[655](379, 1011) = (388, 60)$. In general then, this straightforward double-and-add algorithm will take $\log_2 m$ doublings and roughly half as many additions to compute $[m]P$ (if m is randomly chosen).

2.1.3 The group axioms

All but one of the group axioms are now concrete. Namely, for *closure*, if we start with two points in $E(K)$, then the chord-and-tangent process gives rise to a cubic polynomial in K for which two roots (the two x-coordinates of the points we started with) are in K, meaning the third root must also be in K; the explicit formulas affirm this. The *identity* and *inverse* axioms are fine, since $P \oplus \mathcal{O} = P$, and the element $\ominus P$ such that $P \oplus (\ominus P) = \mathcal{O}$ is clearly unique and well defined for all P. We also note that the group is *abelian*, since the process of computing $P \oplus Q$ is symmetric. The only non-obvious axiom is *associativity*, i.e. showing $(P \oplus Q) \oplus R = P \oplus (Q \oplus R)$. An elementary approach using the explicit formulas above can be used to show associativity by treating all the separate cases, but this approach is rather messy [Fri05]. Silverman gives a much more instructive proof [Sil09, Ch. III.3.4e] using tools that we will develop in the following chapter, but for now we offer some temporary intuition via the illustration in Figures 2.12 and 2.13.

2.1.4 Speeding up elliptic curve computations

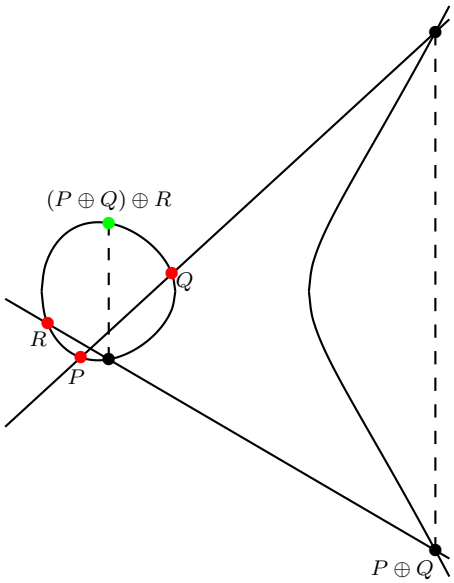


Figure 2.12: $(P \oplus Q) \oplus R$.

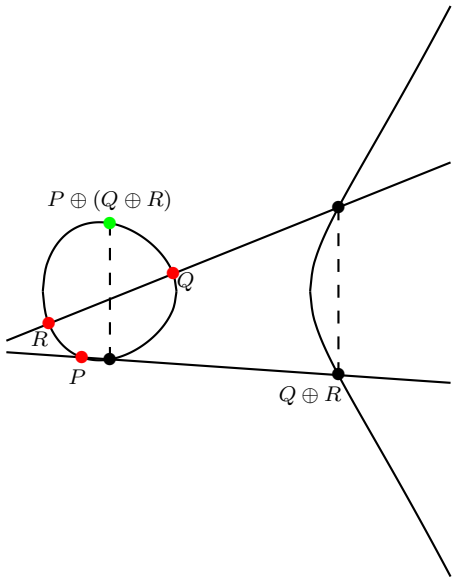


Figure 2.13: $P \oplus (Q \oplus R)$.

Group law computations on elliptic curves are clearly more complicated than computations in traditional groups that facilitate discrete logarithm based protocols like $\mathbb{F}_q^*$; the explicit formulas in (2.4) and (2.5) use many field operations.

However, in the context of cryptography, the more abstract nature of elliptic curve groups actually works in their favour. This is essentially because attackers aiming to solve the discrete logarithm problem on elliptic curves also face this abstractness. The subexponential algorithms that apply to finite field discrete logarithms[1] do not translate to the elliptic curve setting, where the best available attacks remain generic, exponential algorithms like Pollard rho [Pol78]. This means that elliptic curve groups of a relatively small size achieves the same conjectured security as multiplicative groups in much larger finite fields, i.e. $E(\mathbb{F}_{q_1})$ and $\mathbb{F}_{q_2}^*$ achieve similar security when $q_2 \gg q_1$. For example, an elliptic curve defined over a 160-bit field currently offers security comparable to a finite field of 1248 bits [Sma10, Table 7.2]. Thus, although more field operations are required to perform a group law computation, these operations take place in a field whose operational complexity is much less, and this difference is more than enough to tip the balance in the favour of elliptic curves. In addition, the smaller group elements in $E(\mathbb{F}_{q_1})$ implies much smaller key sizes, greatly reducing storage and bandwidth requirements. These are some of the major reasons that elliptic curves have received so much attention in the realm of public-key cryptography; the field of elliptic curve cryptography (ECC) has been thriving since Koblitz [Kob87] and Miller [Mil85] independently suggested their potential as alternatives to traditional groups.

One avenue of research that has given ECC a great boost is that of optimising the group law computations. The explicit formulas in affine coordinates ((2.4) and (2.5)) would not be used to compute the group law in practice, and in fact the Weierstrass model $E : y^2 = x^3 + ax + b$ is often not the optimal curve model either. A huge amount of effort has been put towards investigating other models and coordinate systems in order to minimise the field operations required in group law computations. One of the initial leaps forward in this line of research was the observation that performing computations in projective space avoids field inversions, which are extremely costly in practice. We illustrate these techniques in the following examples.

Example 2.1.9 (Magma script). Consider a general Weierstrass curve $E(\mathbb{F}_q) : y^2 = x^3 + ax + b$ where q is a large prime, and let $\mathbf{M}$, $\mathbf{S}$ and $\mathbf{I}$ represent the cost of computing multiplications, squarings and inversions in $\mathbb{F}_q$ respectively. To compute a general affine point doubling $(x_R, y_R) = [2](x_P, y_P)$ using (2.5) costs

[1] See Diem's notes on *index calculus* for a nice introduction [Die12].

$2\mathbf{M}+2\mathbf{S}+\mathbf{I}$, and to compute a general affine point addition $(x_R, y_R) = (x_P, y_P) \oplus (x_Q, y_Q)$ using (2.4) costs $2\mathbf{M} + \mathbf{S} + \mathbf{I}$. On the other hand, we can transform the formulas into homogeneous projective space according to the substitutions $x = X/Z$ and $y = Y/Z$, and we can consider computing $(X_R : Y_R : Z_R) = [2](X_P : Y_P : Z_P)$ and $(X_R : Y_R : Z_R) = (X_P : Y_P : Z_P) \oplus (X_Q : Y_Q : Z_Q)$ on $E : Y^2Z = X^3 + aXZ^2 + bZ^3$. For the addition case, substituting $x_i = X_i/Z_i$ and $y_i = Y_i/Z_i$ for $i \in \{P, Q, R\}$ into the affine formulas

$$x_R = \left(\frac{y_Q - y_P}{x_Q - x_P}\right)^2 - x_P - x_Q; \qquad y_R = \left(\frac{y_Q - y_P}{x_Q - x_P}\right)(x_P - x_R) - y_P$$

taken from (2.4), gives

$$\frac{X_R}{Z_R} = \left(\frac{\frac{Y_Q}{Z_Q} - \frac{Y_P}{Z_P}}{\frac{X_Q}{Z_Q} - \frac{X_P}{Z_P}}\right)^2 - \frac{X_P}{Z_P} - \frac{X_Q}{Z_Q}; \qquad \frac{Y_R}{Z_R} = \left(\frac{\frac{Y_Q}{Z_Q} - \frac{Y_P}{Z_P}}{\frac{X_Q}{Z_Q} - \frac{X_P}{Z_P}}\right)\left(\frac{X_P}{Z_P} - \frac{X_R}{Z_R}\right) - \frac{Y_P}{Z_P}.$$

After a little manipulation, we can then set Z_R to be the smallest value that contains both denominators above, and update the numerators accordingly to give

$$\begin{aligned}
X_R &= (X_PZ_Q - X_QZ_P)\left(Z_PZ_Q(Y_PZ_Q - Y_QZ_P)^2 - (X_PZ_Q - X_QZ_P)^2(X_PZ_Q + X_QZ_P)\right);\\
Y_R &= Z_PZ_Q(X_QY_P - X_PY_Q)(X_PZ_Q - X_QZ_P)^2\\
&\quad - (Y_PZ_Q - Y_QZ_P)\left((Y_PZ_Q - Y_QZ_P)^2Z_PZ_Q - (X_PZ_Q + X_QZ_P)(X_PZ_Q - X_QZ_P)^2\right);\\
Z_R &= Z_PZ_Q(X_PZ_Q - X_QZ_P)^3.
\end{aligned}$$

The explicit formulas database (EFD) [BL07a] reports that the above formulas can be computed in a total of $12\mathbf{M} + 2\mathbf{S}$. The real power of adopting projective coordinates for computations becomes apparent when we remark that most optimised implementations of $\mathbb{F}_q$ arithmetic have $\mathbf{I} \gg 20\mathbf{M}$, and the multiplication to inversion ratio is commonly reported to be $80 : 1$ or higher. Thus, the $12\mathbf{M} + 2\mathbf{S}$ used for additions in projective space will be much faster than the $2\mathbf{M} + \mathbf{S} + \mathbf{I}$ for affine additions. For completeness, we remark that deriving the projective formulas for computing $(X_R : Y_R : Z_R) = [2](X_P : Y_P : Z_P)$ is analogous (but substantially more compact since we only have the projective coordinates of P to deal with), and the EFD reports that this can be done in $5\mathbf{M}+6\mathbf{S}$, which will again be much faster than the $2\mathbf{M} + 2\mathbf{S} + I$ in affine space.

The Weierstrass model for elliptic curves covers all isomorphism classes, meaning that every elliptic curve can be written in Weierstrass form. Other

models of elliptic curves are usually available if some condition holds, and (if this is the case) it can be advantageous to adopt such a model, as the following example shows.

Example 2.1.10 (Magma script). If $x^3 + ax + b$ has a root in $\mathbb{F}_q$, then Billet and Joye [BJ03, Eq. 8-10] show that instead of working with $E : y^2 = x^3 + ax + b$, we can work with the (birationally equivalent) *Jacobi-quartic* curve $J : v^2 = au^4 + du^2 + 1$, for appropriately defined a, d (that depend on the root). Here we write J using (u, v) coordinates so back-and-forth mappings are defined without confusion. Thus, consider $E/\mathbb{F}_{97} : y^2 = x^3 + 5x + 5$, for which $x^3 + 5x + 5$ has 34 as a root, so we will work on the isomorphic curve $J/\mathbb{F}_{97} : v^2 = 73u^4 + 46u^2 + 1$. Instead of homogeneous projective coordinates, [BJ03] projectified under the substitution $u = U/W$ and $v = V/W^2$, which gives the (non-homogeneous) projective closure as $J : V^2 = 73U^4 + 46U^2W^2 + W^4$. Any point $(x, y) \neq \mathcal{O}$ on E can be taken straight to the projective closure of J via

$$(x, y) \mapsto \left(2(x - 34) : (2x + 34)(x - 34)^2 - y^2 : y\right),$$

with the reverse mapping given by

$$(U : V : W) \mapsto \left(2\frac{V + W^2}{U^2} - 17, W\frac{4(V + W^2) - 5U^2}{U^3}\right).$$

For example $(x, y) = (77, 21)$ maps to $(U : V : W) = (86 : 8 : 21)$, and vice versa. We now look at the formulas for the point addition $(U_3 : V_3 : W_3) = (U_1 : V_1 : W_1) \oplus (U_2 : V_2 : W_2)$ on $J : V^2 = aU^4 + dU^2W^2 + W^4$, taken from [BJ03, Eq. 11], as

$$\begin{aligned}
U_3 &= U_1W_1V_2 + U_2W_2V_1, \\
V_3 &= \left((W_1W_2)^2 + a(U_1U_2)^2\right)(V_1V_2 + dU_1U_2W_1W_2) + 2aU_1U_2W_1W_2(U_1^2W_2^2 + U_2^2W_1^2), \\
W_3 &= (W_1W_2)^2 - a(U_1U_2)^2,
\end{aligned}$$

where we immediately highlight the relative simplicity of the above formulas in comparison to the homogeneous projective formulas derived in the previous example. Unsurprisingly then, the fastest formulas for Jacobi-quartic additions and doublings outdo those for general Weierstrass curves in homogeneous projective space. Namely, the current fastest formulas for doublings on Jacobi-quartics cost $2\mathbf{M} + 5\mathbf{S}$ and additions cost $6\mathbf{M} + 4\mathbf{S}$ [HWCD09], whilst in the previous

example we had $5\mathbf{M} + 6\mathbf{S}$ for doublings and $12\mathbf{M} + 2\mathbf{S}$ for additions.

The Jacobi-quartic curves discussed above are just one example of dozens of models that have been successful in achieving fast group law computations, and therefore fast cryptographic implementations. Other well known models include Edwards curves [Edw07, BL07b], Hessian curves [JQ01, Sma01] and Montgomery curves [Mon87]. We refer to the EFD [BL07a] for a catalogue of all the fastest formulas for the popular curve models, and to Hisil's thesis [His10] for a general method of (automatically) deriving fast group law algorithms on arbitrary curve models. For any reader wishing to delve even further into group law arithmetic on elliptic curves, we also recommend the recent, advanced works by Castryck and Vercauteren [CV11], and by Kohel [Koh11].

2.2 Torsion, endomorphisms and point counting

We now turn our focus to the behaviour of elliptic curve groups, as they are used in cryptography. We start by importantly discussing the possible structures exhibited by the finite group $E(\mathbb{F}_q)$. It turns out that $E(\mathbb{F}_q)$ is either itself cyclic, or isomorphic to a product of two cyclic groups $\mathbb{Z}_{n_1} \times \mathbb{Z}_{n_2}$ with $n_1 \mid n_2$ [ACD+05, Prop. 5.78]. In cryptography, we would like the group $E(\mathbb{F}_q)$ to be *as cyclic as possible*, so we usually prefer the former case, or at the very least for n_1 to be very small. In most cases of practical interest, we can generate curves that are cyclic with relative ease, so throughout this thesis it is to safe assume that $E(\mathbb{F}_q)$ is cyclic (but to see the real depth of this question in general, we refer to [MS07]). The following example illustrates that $E(\mathbb{F}_q) = \langle P \rangle$ obeys all the usual rules that apply to cyclic groups, and introduces the important notion of *r-torsion*.

Example 2.2.1 (Magma script). Consider $E/\mathbb{F}_{101} : y^2 = x^3 + x + 1$. The group order is $\#E(\mathbb{F}_q) = 105 = 3 \cdot 5 \cdot 7$, and $P = (47, 12) \in E$ is a generator. Lagrange's theorem says that points (and subgroups) over the base field will have order in $\{1, 3, 5, 7, 15, 21, 35, 105\}$. Indeed, to get a point of order $r \mid 105$, we simply multiply P by the appropriate *cofactor*, which is $h = \#E/r$. For example, a point of order 3 is $[35](47, 12) = (28, 8)$, a point of order 21 is $[5](47, 12) = (55, 65)$, and a point of order 1 is $[105](47, 12) = \mathcal{O}$ (which is the only such point). By

definition, a point is "killed" (sent to $\mathcal{O}$) when multiplied by its order. Any point over the full closure $E(\overline{\mathbb{F}}_q)$ that is killed by r is said to be in the r-torsion. So, the point $(55, 65)$ above is in the 21-torsion, as is the point $(28, 8)$. There are exactly 21 points in $E(\mathbb{F}_q)$ in the 21-torsion, but there are many more in $E(\overline{\mathbb{F}}_q)$.

The whereabouts and structure of r-torsion points in $E(\overline{\mathbb{F}}_q)$ (alluded to at the end of Example 2.2.1) plays a crucial role in pairing-based cryptography; we will be looking at this in close detail in Chapter 4.

In ECC we would like the group order $\#E(\mathbb{F}_q)$ to be as close to prime as possible. This is because the (asymptotic) complexity of the ECDLP that attackers face is dependent on the size of the largest prime subgroup of $E(\mathbb{F}_q)$. Even if the particular instance of the discrete logarithm problem uses a generator of the whole group, the attacker can use the known group order to solve smaller instances in subgroups whose orders are pairwise prime, and then reconstruct the answer using the Chinese Remainder Theorem (CRT). We make this clear in the following two examples: the first is a toy example, whilst the second shows the difference between two curves of the same cryptographic size; one that is currently considered secure and one that is completely breakable using modern attacks.

Example 2.2.2 (Magma script). Consider $E/\mathbb{F}_{1021} : y^2 = x^3 + 905x + 100$, with group order $\#E(\mathbb{F}_q) = 966 = 2 \cdot 3 \cdot 7 \cdot 23$, and generator $P = (1006, 416)$. Suppose we are presented with an instance of the ECDLP: namely, we are given $Q = (612, 827)$, and we seek to find k such that $[k]P = Q$. For the sake of the example, suppose our best "attack" is trivial: trying every multiple $[i]P$ of P until we hit the correct one ($i = k$). Rather than seeking i in the full group ($2 \leq i \leq 965$), we can map the instance into each prime order subgroup by multiplying by the appropriate cofactor, and then solve for $k_j \equiv k \bmod j$, $j \in \{2, 3, 7, 23\}$. For $j = 2$, we have $P_j = P_2 = [966/2]P = [483](1006, 416) = (174, 0)$, and $Q_j = Q_2 = [483](612, 827) = (174, 0)$, so $Q_2 = [k_2]P_2$ gives $k_2 = 1$. For $j = 3$, we have $P_3 = [322]P = (147, 933)$ and $Q_3 = [322]P = \mathcal{O}$, so $Q_3 = [k_3]P_3$ gives $k_3 = 3$. For $j = 7$, we have $P_7 = [138]P = (906, 201)$ and $Q_7 = [138]Q = (906, 201)$, so $Q_7 = [k_7]P_7$ gives $k_7 = 1$. For $j = 23$, we have $P_{23} = [42]P = (890, 665)$ and $Q_{23} = [42]Q = (68, 281)$. For $Q_{23} = [k_{23}]P_{23}$, we exhaust $k_{23} \in \{1, .., 22\}$ to see that $k_{23} = 20$. Now, we can use the Chinese Remainder Theorem to solve

$$k \equiv k_2 = 1 \bmod 2; \quad k \equiv k_3 = 0 \bmod 3; \quad k \equiv k_7 = 1 \bmod 7; \quad k \equiv k_{23} = 20 \bmod 23,$$

which gives $k \equiv 687 \bmod \#E$, solving the ECDLP instance. Notice that the

hardest part was exhausting the set $\{1, .., 22\}$ to find $k_{23} = 20$, so the largest prime order subgroup becomes the bottleneck of the algorithm, giving intuition as to why the largest prime order subgroup defines the attack complexity when groups of a cryptographic size are used.

Example 2.2.3 (Magma script). For our real world example, we take the curve P-256 from the NIST recommendations [NIS99], which currently achieves a similar security level (resistance against best known attacks) to the 128-bit Advanced Encryption Standard (AES) for symmetric encryption. The curve is defined as $E/\mathbb{F}_q : y^2 = x^3 - 3x + b$, with prime order $r = \#E$, and generator $G = (x_G, y_G)$, where

$q = 115792089210356248762697446949407573530086143415290314195533631308867097853951,$
$r = 115792089210356248762697446949407573529996955224135760342422259061068512044369,$
$b = 41058363725152142129326129780047268409114441015993725554835256314039467401291,$
$x_G = 48439561293906451759052585252797914202762949526041747995844080717082404635286,$
$y_G = 36134250956749795798585127919587881956611106672985015071877198253568414405109,$
$x_H = 53987601597021778433910548064987973235945515666715026302948657055639179420355,$
$y_H = 53690949263410447908824456005055253553237881490194075871737490561466076234637.$

We give another point $H = (x_H, y_H)$ to pose $H = [k]G$ as an intractable instance of the ECDLP; this 256-bit prime field (and group order) is far beyond the reach of current attacks. For example, there is currently a campaign underway to solve a discrete logarithm problem over a 130-bit field using a cluster of servers that have already been running for two years (see `http://ecc-challenge.info/`), so (assuming the best known attacks stay exponential) it seems the above ECDLP should be safe for a while yet. We remark that the prime characteristic q is given by $q = 2^{256} - 2^{224} + 2^{192} + 2^{96} - 1$; such primes are preferred in ECC as they allow for faster finite field multiplication and reduction routines, greatly enhancing the speed of $\mathbb{F}_q$ arithmetic. We now give a curve over the same field $\mathbb{F}_q$, for which the ECDLP is well within reach of the best known attacks. Namely, consider the alternative curve with $b = 0$, namely $\tilde{E}/\mathbb{F}_q : y^2 = x^3 - 3x$, whose group order $n = \#\tilde{E}$ is given as

$$n = 115792089210356248762697446949407573530086143415290314195533631308867097853952,$$
$$= 2^{96} \cdot 7 \cdot 274177 \cdot 67280421310721 \cdot 1131830892797394193140491410 3.$$

This time, the largest prime divisor of the group order is only 94 bits long, and the complexity of solving the ECDLP in $\tilde{E}(\mathbb{F}_q)$ is governed by the difficulty of solving the ECDLP instance in this largest prime subgroup, which could be done in a small amount of time on a desktop computer.

The above example provides clear motivation as to the importance of counting points on elliptic curves. The largest prime factor of the group order determines the difficulty that attackers face when trying to solve the ECDLP, so we would like to be able to count points on curves quickly enough to find those whose order is prime or almost prime (i.e. has a small cofactor), or have methods of prescribing such a group order before searching for the curve. Fortunately, on elliptic curves we have efficient algorithms to do both.

We start our brief discussion on elliptic curve point counting by referring back to the two group orders in Example 2.2.3, and observing that both group orders share the first half of their digits with those of the field characteristic q. This suggests that the number of points on an elliptic curve is close to q, which is indeed the case in general; the *Hasse bound* [Sil09, Ch. 5, Th. 1.1] says the most that $\#E(\mathbb{F}_q)$ can differ from $q+1$ is $2\sqrt{q}$, i.e. $|\#E(\mathbb{F}_q) - (q+1)| \leq 2\sqrt{q}$. This offset between $\#E(\mathbb{F}_q)$ and $(q+1)$ is called the *trace of Frobenius*, and is denoted by t, so

$$\#E(\mathbb{F}_q) = q + 1 - t, \qquad |t| \leq 2\sqrt{q} \tag{2.6}$$

We will discuss where t comes from and provide some more intuition behind the above formula in a moment, but what the Hasse bound tells us is that the group order lies somewhere in the interval $[q+1-2\sqrt{q}, q+1+2\sqrt{q}]$. In fact, Deuring [Deu41] showed that when q is prime[2], then every value $N \in [q+1-2\sqrt{q}, q+1+2\sqrt{q}]$ can be found as a group order $\#E(\mathbb{F}_q)$ for some E.

Example 2.2.4 (Magma script). Let $q = 23$, so that the Hasse interval becomes $[q+1-2\sqrt{q}, q+1+2\sqrt{q}] = [15, 33]$, meaning that there are exactly 19 different

[2]When q is a prime power, there are a very small number of explicitly described exceptions.

group orders taken by elliptic curves over $\mathbb{F}_{23}$. For example, $E/\mathbb{F}_{23} : y^2 = x^3 + 18x + 3$ has $\#E = 15$, whilst $\tilde{E}/\mathbb{F}_{23} : y^2 = x^3 + 13x + 7$ has $\#\tilde{E} = 33$. We give 19 (a, b) pairs such that the corresponding curves $E : y^2 = x^3 + ax + b$ have group orders in ascending order spanning the whole interval, as follows: $(18, 3)$, $(7, 22)$, $(19, 14)$, $(17, 17)$, $(12, 5)$, $(7, 12)$, $(8, 10)$, $(17, 18)$, $(20, 20)$, $(2, 3)$, $(20, 3)$, $(6, 8)$, $(16, 8)$, $(16, 22)$, $(9, 16)$, $(19, 6)$, $(20, 8)$, $(22, 9)$, $(13, 7)$.

A rough (but elementary and instinctive) argument as to why $\#E \approx q$ is that approximately half of the values $x \in [0, .., q-1]$ will give a quadratic residue $x^3 + ax + b \in \mathrm{QR}(q)$, which gives rise to two points $(x, \pm\sqrt{x^3 + ax + b}) \in E(\mathbb{F}_q)$, the only exception(s) being when $x^3 + ax + b = 0$ which obtains one point. The sophisticated explanation requires a deeper knowledge than our introduction offers, but for the purposes of this introductory text we get almost all that we need from Equation (2.6); the derivation of which makes use of the following definition. If E is defined over $\mathbb{F}_q$, then the *Frobenius endomorphism* π is defined as

$$\pi : E \to E, \qquad (x, y) \mapsto (x^q, y^q). \tag{2.7}$$

We note that the Frobenius endomorphism maps any point in $E(\overline{\mathbb{F}}_q)$ to a point in $E(\overline{\mathbb{F}}_q)$, but the set of points fixed by π is exactly the group $E(\mathbb{F}_q)$. Thus, π only acts non-trivially on points in $E(\overline{\mathbb{F}}_q) \setminus E(\mathbb{F}_q)$, and more generally, $\pi^i : (x, y) \mapsto (x^{q^i}, y^{q^i})$ only acts non-trivially on points in $E(\overline{\mathbb{F}}_q) \setminus E(\mathbb{F}_{q^i})$.

Example 2.2.5 (Magma script). Let $q = 67$, and consider $E/\mathbb{F}_q : y^2 = x^3 + 4x + 3$, and let $\mathbb{F}_{q^2} = \mathbb{F}_q(u)$ where $u^2 + 1 = 0$, and further let $\mathbb{F}_{q^3} = \mathbb{F}_q(v)$ where $v^3 + 2 = 0$. For $P_1 = (15, 50) \in E(\mathbb{F}_q)$, we have $\pi_q(P_1) = (15^q, 50^q) = (15, 50)$. For $P_2 = (2u + 16, 30u + 39)$, we have $\pi_q(P_2) = ((2u + 16)^q, (30u + 39)^q) = (65u + 16, 39 + 37u)$; it is easy to see in this example that computing $\pi_q(Q)$ for any $Q \in E(\mathbb{F}_{q^2})$ involves a simple "complex conjugation" on each coordinate, which also agrees with $\pi_q^2(Q) = Q$. Let $P_3 = (15v^2 + 4v + 8, 44v^2 + 30v + 21)$, $\pi_q(P_3) = (33v^2 + 14v + 8, 3v^2 + 38v + 21)$, $\pi_q^2(P_3) = (19v^2 + 49v + 8, 20v^2 + 66v + 21)$, and $\pi_q^3(P_3) = P_3$.

We can now return to sketch the derivation of Equation (2.6) by skimming over results that are presented in full in Silverman's book [Sil09, Ch. V, Th. 1.1]. We now know that $P \in E(\mathbb{F}_q)$ if and only if $\pi(P) = P$ (i.e. $([1] - \pi)P = \mathcal{O}$), and thus $\#E(\mathbb{F}_q) = \#\ker([1] - \pi)$. It is not too hard to show that the map

$[1] - \pi$ is separable, which means that $\#E(\mathbb{F}_q) = \#\ker([1] - \pi) = \deg([1] - \pi)$. We can then make use of (a special case of) a version of the Cauchy-Schwarz inequality [Sil09][Ch. V, Lemma 1.2], to give $|\deg([1] - \pi) - \deg([1]) - \deg(\pi)| \leq 2\sqrt{\deg([1])\deg(\pi)}$, from which Equation (2.6) follows from $\deg(\pi) = q$.

The theory of elliptic curves makes constant use of the *endomorphism ring* of E, denoted $\mathrm{End}(E)$, which (as the name suggests) is the ring of all maps from E to itself; addition in the ring is natural, i.e. $(\psi_1 + \psi_2)(P) = \psi_1(P) + \psi_2(P)$, and multiplication in $\mathrm{End}(E)$ is composition $(\psi_1\psi_2)(P) = \psi_1(\psi_2(P))$. The *multiplication-by-m* map $[m]$ is trivially in $\mathrm{End}(E)$ for all $m \in \mathbb{Z}$, and when E is defined over a finite field, then clearly π is too, so we are usually interested in any extra endomorphisms that shed more light on the behaviour of E.

Example 2.2.6 (Magma script). Consider $E/\mathbb{F}_q : y^2 = x^3 + b$. The map ξ, defined by $\xi : (x, y) \mapsto (\xi_3 x, y)$ with $\xi_3^3 = 1$ and $\xi_3 \neq 1$, is a non-trivial endomorphism on E, so $\xi \in \mathrm{End}(E)$. If $\xi_3 \in \mathbb{F}_q$, then ξ will be defined over $\mathbb{F}_q$, otherwise $\xi_3 \in \mathbb{F}_{q^2}$ in which case ξ is not *defined over* $\mathbb{F}_q$, but over $\mathbb{F}_{q^2}$. We will observe both cases. Firstly, cubic roots of unity will be defined in $\mathbb{F}_q$ if and only if $q \equiv 1 \bmod 3$, so let us take $q \equiv 19$, $b = 5$, which gives $E/\mathbb{F}_{19} : y^2 = x^3 + 5$. Let $\xi_3 = 7$ so that $\xi_3^3 = 1$ (we could have also taken $\xi_3^2 = 11$), so that $\xi : (x, y) \mapsto (7x, y)$ is an endomorphism on E. Applying this to, say $P = (-1, 2)$, gives $\xi(P) = (-7, 2) \in E$. Taking the same curve over $\mathbb{F}_{23}$, i.e. $E/\mathbb{F}_{23} : y^2 = x^3 + 5$, for which $P = (-1, 2)$ is a again a point, we no longer have a non-trivial $\xi_3 \in \mathbb{F}_{23}$, so we must form a quadratic extension $\mathbb{F}_{q^2}(u)$, $u^2 + 1 = 0$. Now, we can take $\xi_3 = 8u + 11$ (the other option is $\xi_3^2 = 15u + 11$), so that $\xi(P) = (-(8u + 11), 2) = (15u + 12, 2) \in E(\mathbb{F}_{q^2})$. Notice that P started in $E(\mathbb{F}_q)$, but landed in $E(\mathbb{F}_{q^2})$ under ξ. The endomorphism ξ has an inverse ξ^{-1} (which is defined the same way but with ξ_3^2 instead), so ξ is actually an automorphism of E, written as $\xi \in \mathrm{Aut}(E)$.

The definition of $\xi : (x, y) \mapsto (\xi_3 x, y)$ in the above example gives an endomorphism on $E : y^2 = x^3 + b$ regardless of the field that E is defined over. If there exists a non-trivial map (like ξ) for an elliptic curve E, we say E has *complex multiplication*. To be more precise, all elliptic curve endomorphism rings trivially contain $\mathbb{Z}$, since every $m \in \mathbb{Z}$ corresponds to the multiplication-by-m map $[m] \in \mathrm{End}(E)$. However, if non-trivial endomorphisms exist that make $\mathrm{End}(E)$ strictly larger than $\mathbb{Z}$, then we say E has complex multiplication (CM). Thus, by this definition, every elliptic curve defined over $\mathbb{F}_q$ has CM, because the existence of the Frobenius endomorphism $\pi \in \mathrm{End}(E)$ makes $\mathrm{End}(E)$ larger than $\mathbb{Z}$.

However, if we discuss whether E has CM without yet stipulating the underlying finite field, then the question becomes non-trivial in general, because the answer depends on the existence of non-trivial maps. We use Silverman's example to illustrate [Sil09, Ch. 3, Eg. 4.4].

Example 2.2.7 (Magma script). Consider $E/K : y^2 = x^3 + ax$. The map $\zeta : (x, y) \mapsto (-x, iy)$, where $i^2 = -1$ in K is an endomorphism, so E has CM. Clearly, ζ will be defined over K if and only if $i \in K$. Observe that $\zeta \circ \zeta(x, y) = \zeta(-x, iy) = (x, -y) = -(x, y)$, so $\zeta \circ \zeta = [-1]$ (i.e. ζ^2 is equivalent to negation). Thus, there is a ring homomorphism $\mathbb{Z}[i] \to \mathrm{End}(E)$ defined by $m + ni \mapsto [m] + [n] \circ \zeta$. If $\mathrm{Char}(K) \neq 0$, then this map is an isomorphism, thus $\mathrm{End}(E) \cong \mathbb{Z}[i]$, and $\mathrm{Aut}(E) \cong \mathbb{Z}[i]^*$.

The trace of Frobenius t in Equation (2.6) is named so because of the role it plays in the characteristic polynomial satisfied by π, which is given as

$$\pi^2 - [t] \circ \pi + [q] = 0 \qquad \text{in } \mathrm{End}(E), \tag{2.8}$$

meaning that for all $(x, y) \in E(\overline{\mathbb{F}}_q)$, we have

$$(x^{q^2}, y^{q^2}) - [t](x^q, y^q) + [q](x, y) = \mathcal{O}. \tag{2.9}$$

Example 2.2.8 (Magma script). We use our results from Example 2.2.5 to illustrate, so as before $E/\mathbb{F}_{67} : y^2 = x^3 + 4x + 3$, $\mathbb{F}_{q^2} = \mathbb{F}_q(u)$ where $u^2 + 1 = 0$, and $\mathbb{F}_{q^3} = \mathbb{F}_q(v)$ where $v^3 + 2 = 0$. The trace of Frobenius is $t = -11$, so $\#E(\mathbb{F}_q) = q + 1 - t = 79$. For $P_1 = (15, 50) \in E(\mathbb{F}_q)$, we trivially had $\pi^2(P_1) = \pi(P_1) = P_1$, so $P_1 - [t]P_1 + [q]P_1 = ([1] - [t] + [q])P_1 = [\#E(\mathbb{F}_q)]P_1 = \mathcal{O}$. For $P_2 = (2u+16, 30u+39)$, we had $\pi^2(P_2) = P_2$ and $\pi(P_2) = (65u+16, 37u+39)$, so we are computing $P_2 - [-11]\pi(P_2) + [67]P_2 = [68](2u+16, 30u+39) + [11](65u+16, 37u+39)$, which is indeed $\mathcal{O}$. $P_3 \in E(\mathbb{F}_{q^3})$ is the only case where both π and π^2 act non-trivially, so we compute $(19v^2+49v+8, 20v^2+66v+21) - [-11](33v^2+14v+8, 3v^2+38v+21) + [67](15v^2+4v+8, 44v^2+30v+21)$, which is $\mathcal{O}$.

We now give a brief sketch of Schoof's algorithm for counting points on elliptic curves [Sch85]. Understanding the algorithm is not a prerequisite for understanding pairings, but it certainly warrants mention in any overview text on elliptic curves in cryptography, since it is essentially the algorithm that made ECC practical. Before Schoof's polynomial-time algorithm, all algorithms for point counting on elliptic curves were exponential and therefore cryptographi-

cally impractical. Besides, to sketch his idea, we need to introduce the notion of *division polynomials*, which are a useful tool in general. Put simply, division polynomials are polynomials whose roots reveal torsion points: namely, for odd[3] ℓ, the ℓ-th division polynomial $\psi_\ell(x)$ on E solves to give the x-coordinates of the points of order ℓ. They are defined recursively and depend on the curve constants a and b, but rather than giving the recursions here, we point the reader to [Sil09, Ch. III, Exer. 3.7], and opt instead for an example that illustrates their usefulness.

Example 2.2.9 (Magma script). Recall the curve $E/\mathbb{F}_{101} : y^2 = x^3 + x + 1$ from Example 2.2.1 with group order $\#E(\mathbb{F}_q) = 105 = 3 \cdot 5 \cdot 7$. The x-coordinates of the points of order 2 are found as the roots of $\psi_2(x) = 4x^3 + 4x + 4$, which is irreducible in $\mathbb{F}_q[x]$, so there are no 2-torsion points in $E(\mathbb{F}_q)$. For $r = 3$, $\psi_3(x) = 3x^4+6x^2+12x+100 \in \mathbb{F}_q[x]$ factors into $\psi_3(x) = (x+73)(x+84)(x^2+45x+36)$, so we get two solutions over $\mathbb{F}_q$, namely $x = 17$ and $x = 28$. This does not mean that the points implied by both solutions are in $\mathbb{F}_q$: namely, $x = 28$ gives $x^3 + x + 1 \in \mathrm{QR}(q)$, so two points in the 3-torsion follow as $(28, 8)$ and $(28, 93)$. Conversely, $x = 17$ gives $x^3+x+1 \notin \mathrm{QR}(q)$, so the two points implied by $x = 17$ will be defined over $\mathbb{F}_{q^2}$. For $\psi_5(x) = 5x^{12} + ... + 16$, the factorisation in $\mathbb{F}_q[x]$ is $\psi_5(x) = (x + 15)(x + 55)(x^5 + ... + 1)(x^5 + ... + 100)$, which gives $x = 46$ and $x = 86$ as solutions. This time, both x values give rise to two points, giving four non-trivial 5-torsion points in total: $(46, 25)$, $(46, 76)$, $(86, 34)$, $(86, 67)$. $\psi_7(x)$ is degree 24, and gives three linear factors in $\mathbb{F}_q[x]$, all of which result in two 7-torsion points, giving 6 non-trivial torsion points in total: $(72, 5)$, $(72, 96)$, $(57, 57)$, $(57, 44)$, $(3, 43)$, $(3, 58)$. Other division polynomials have roots in $\mathbb{F}_q$, but these roots will not give rise to points defined over $\mathbb{F}_q$. For example, $\psi_{11}(x)$ has 5 roots over $\mathbb{F}_q$ (13, 18, 19, 22, 63), but none of them give points in $E(\mathbb{F}_q)$, meaning we will have to extend to $E(\mathbb{F}_{q^2})$ to collect any 11-torsion points. The only division polynomials whose roots produce points defined over $\mathbb{F}_q$ are the $\psi_d(x)$ with $d \mid 105$. This generalises to imply that the only division polynomials whose roots produce points defined over $\mathbb{F}_{q^n}$ are $\psi_d(x)$, where $d \mid \#E(\mathbb{F}_{q^n})$.

We are now in a position to shed light on Schoof's algorithm. Equation (2.6) means that computing $E(\mathbb{F}_q)$ immediately reduces to computing the (much smaller) trace of Frobenius, t. At the highest level, Schoof's idea is to compute

[3]When ℓ is even, the division polynomial is of the form $\psi_\ell(x, y) = y \cdot \tilde{\psi}_\ell(x)$ since $y = 0$ gives points of order two, which are in the ℓ-torsion.

$t_\ell \equiv t \bmod \ell$ for enough co-prime ℓ's to be able to uniquely determine t within the interval $-2\sqrt{q} \leq t \leq 2\sqrt{q}$ via the Chinese Remainder Theorem. Namely, when $\prod_\ell t_\ell \geq 4\sqrt{q}$, then we have enough relations to determine the correct t. To compute t_ℓ for various primes ℓ, Schoof looked to consider Equation (2.9) "modulo ℓ", restricting the points (x, y) to come from the ℓ-torsion, and trying to solve

$$(x^{q^2}, y^{q^2}) - [t_\ell](x^q, y^q) + [q_\ell](x, y) = \mathcal{O}, \tag{2.10}$$

for t_ℓ, where $q_\ell \equiv q \bmod \ell$. The problem for general ℓ is, that since we do not know the group order, we cannot explicitly use ℓ-torsion points in (2.10), nor do we know if they are even defined over $\mathbb{F}_q$, or where they *are* defined, so we have to work with (2.10) implicitly. Namely, we restrict (2.10) to the ℓ-torsion by working modulo $\psi_\ell(x)$: we do not work with Equation (2.10) on $E(\mathbb{F}_q)$, but rather in the polynomial ring $R_\ell = \mathbb{F}_q[x, y]/\langle \psi_\ell(x), y^2 - (x^3 + ax + b)\rangle$, where the size of the polynomials $f(x, y)$ we deal with in R_ℓ are bounded by the degrees of the division polynomials $\psi_\ell(x)$. Even for very large prime fields $\mathbb{F}_q$ of cryptographic size, the number of different primes used is small enough to keep this algorithm very practical. For example, finding the group order of the curve defined over a 256-bit prime q in Example 2.2.3 would require solving (2.10) for the 27 primes up to $\ell = 107$, at which point the product of all the primes used exceeds $4\sqrt{q}$. It is not too difficult to deduce that the asymptotic complexity of Schoof's algorithm is $O\left((\log q)^8\right)$ (see [Sil09, Ch. XI.3] for details, and further improvements).

Example 2.2.10 (Magma script). Consider $E/\mathbb{F}_{13} : y^2 = x^3 + 2x + 1$; we seek $\#E(\mathbb{F}_{13})$. Schoof's algorithm actually begins with $\ell = 3$ [Sil09, Ch. XI.3]; so since $14 < 4\sqrt{13} < 15$, we only need to solve (2.10) with $\ell = 3$ and $\ell = 5$. For $\ell = 3$, $\psi_3(x) = 3x^4 + 12x^2 + 12x + 9$, so we work in the ring $R_3 = \mathbb{F}_q[x, y]/\langle 3x^4 + 12x^2 + 12x + 9, y^2 - (x^3 + 2x + 1)\rangle$ with $q_\ell = 1$, to find that $t_3 = 0$. For $\ell = 5$, $\psi_5(x) = 5x^{12} + ... + 6x + 7$, so we work in the ring $R_5 = \mathbb{F}_q[x, y]/\langle 5x^{12} + ... + 6x + 7, y^2 - (x^3 + 2x + 1)\rangle$ with $q_\ell = 3$ to find that $t_5 = 1$. For both cases we had to compute $[q_\ell](x, y)$ in R_ℓ using the affine formulas (2.4) and (2.5), compute (x^q, y^q) and (x^{q^2}, y^{q^2}) in R_ℓ, and then test incremental values of t_ℓ until $[t_\ell](x^q, y^q)$ (also computed with the affine formulas) satisfies (2.10). The CRT with $t \equiv 0 \bmod 3$ and $t \equiv 1 \bmod 5$ gives $t \equiv 6 \bmod 15$, which combined with $-7 \leq t \leq 7$ means $t = 6$, giving $\#E = q + 1 - t = 8$.

We finish this chapter by briefly discussing one more improvement to ECC

that will essentially bring the reader up to speed with major milestones that contribute to the current state-of-the-art implementations. The technique was introduced by Gallant, Lambert and Vanstone (GLV) [GLV01], and recently generalised by Galbraith, Lin and Scott (GLS) [GLS11]. It exploits the existence of an efficiently computable endomorphism ψ that allows us to instantly move P to a large multiple $\psi(P) = [\lambda]P$ of itself, so that (in the simplest case) the scalar multiplication $[m]P$ can be split into $[m]P = [m_0]P + [m_1]\psi(P)$, where if $|m| \approx r$ (the large subgroup order), then $|m_0|$, $|m_1| \approx \sqrt{r}$. The values m_0 and m_1 are found by solving a closest vector problem in a lattice [GLV01, §4]. We apply an example from the GLV paper (which was itself taken from Cohen's book [Coh96, §7.2.3]) that is actually exploiting a special case of the endomorphism we described in Example 2.2.7.

Example 2.2.11 (Magma script). Let $q \equiv 1 \bmod 4$ be prime, $E/\mathbb{F}_q : y^2 = x^3 + ax$, and let $i^2 = -1$. The map defined by $\psi : (x, y) \mapsto (-x, iy)$ and $\psi : \mathcal{O} \mapsto \mathcal{O}$ is an endomorphism defined over $\mathbb{F}_q$ ($\psi = \zeta$ from 2.2.7). Let $P \in E(\mathbb{F}_q)$ have prime order r, then $\psi(Q) = [\lambda]Q$ for all $Q \in \langle P \rangle$, and λ is the integer satisfying $\lambda^2 = -1 \bmod r$. We give a specific example: $q = 1048589$, $E/\mathbb{F}_q : y^2 = x^3 + 2x$ with $\#E = 2r$, where $r = 524053$; we further have $i = 38993$, and $\lambda = 304425$. $P = (609782, 274272) \in E$ has $|\langle P \rangle| = r$, so we can take any element in $\langle P \rangle$, say $Q = (447259, 319154)$, and compute $\psi(Q) = (-447259, i \cdot 319154) = (601330, 117670) = [304425](447259, 319154) = [\lambda]Q$. Computing a random multiple of Q, say $[m]Q$ with $m = 103803$, can be done by decomposing m into (in this case) $(m_0, m_1) = (509, 262)$, and instead computing $[m]Q = [m_0]Q + [m_1]\psi(Q)$. Here m is 17 bits, whilst m_0 and m_1 are both 9 bits. Doing the scalar multiples $[m_0]Q$ and $[m_1]\psi(Q)$ separately would therefore give no savings, but where the GLV/GLS methods gain a substantial speed-up is in merging the doublings required in both of the multiplications by the "mini-scalars", which halves the number of doublings required overall; again, see [GLV01, GLS11] for futher details.

2.3 Chapter summary

We defined the elliptic curve group law $\oplus$ via the chord-and-tangent method, and discussed that elliptic curve groups are an attractive setting for discrete-log based cryptosystems because of the relative security obtained for the sizes of the

fields they are defined over. We also exemplified many improvements in the context of cryptographic implementations, where the fundamental operation (that creates ECDLP instances) is computing large scalar multiples $[m]P$ of $P \in E$. Namely, we showed that group law computations in finite fields can be much faster in projective coordinates, i.e. computing $(X_1 : Y_1 : Z_1) \oplus (X_2 : Y_2 : Z_2)$ rather than $(x_1, y_1) \oplus (x_2, y_2)$, and that other (non-Weierstrass) curve models also offer advantages. We gave an explicit equation for the number of points in $E(\mathbb{F}_q)$, and briefly discussed Schoof's polynomial-time algorithm that facilitates point counting on curves of cryptographic size. We also introduced the notion of the endomorphism ring $\mathrm{End}(E)$ of E, and finished by showing that non-trivial elements of $\mathrm{End}(E)$ can be used to further accelerate ECC. A reader that is comfortable with the exposition in this chapter is equipped with many of the tools required to tackle the vast literature in this field, and is somewhat up-to-date with the state-of-the-art ECC implementations. For example, in the context of chasing ECC speed records, some authors have applied alternative projective coordinate systems to the Edwards model to give very fast scalar multiplications [HWCD08], whilst others have investigated higher dimension GLV/GLS techniques (Example 2.2.11 above was 2-dimensional) to gain big speed-ups [HLX12]; visit `http://bench.cr.yp.to/supercop.html` for comprehensive and up-to-date benchmarkings of a wide number of implementations that are pushing ECC primitives to the limit.

Relaxed notation. Our last order of business before proceeding into the next chapter is to relax some notation in order to agree with the rest of the literature. Rather than writing "$\oplus$" for the elliptic curve group law, from hereon we simply use "$+$". Similarly, for the inverse of the point P, we use $-P$ instead of $\ominus P$.