

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

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

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

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

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.

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

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

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 $x < 6$.

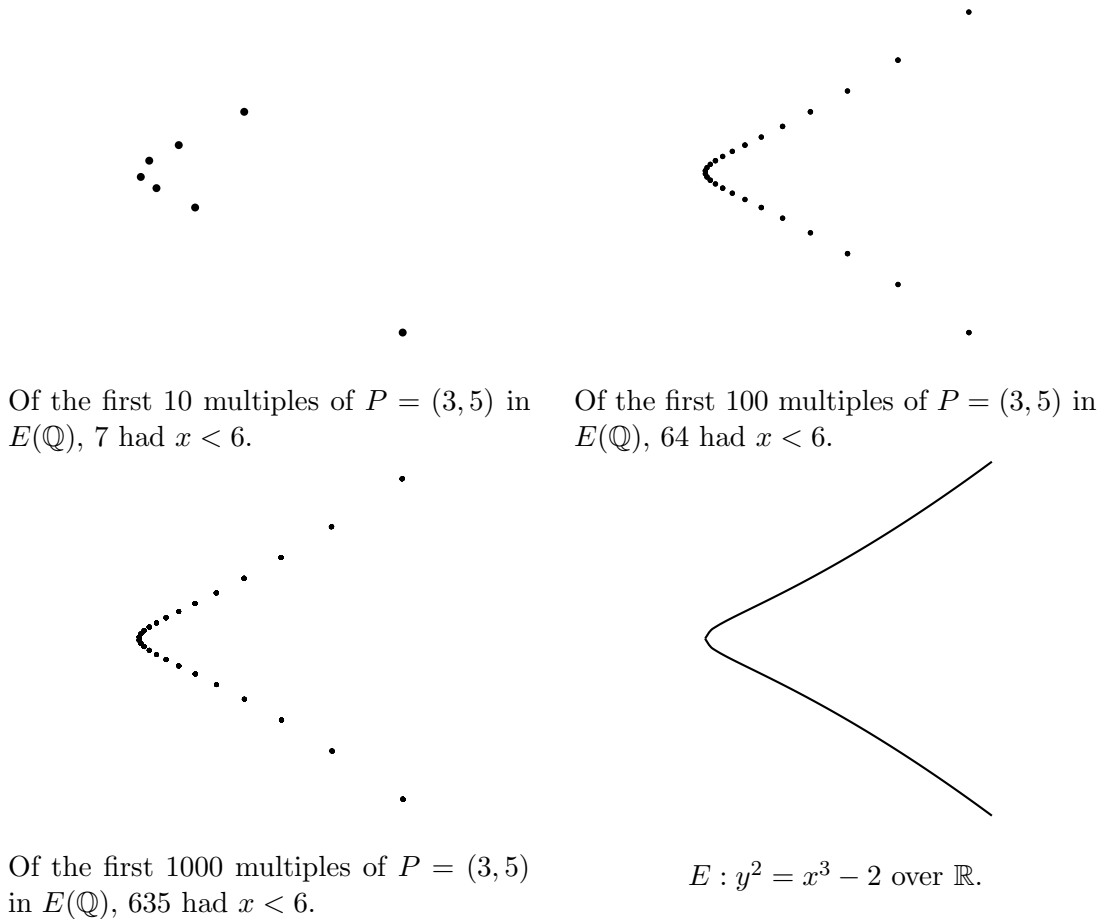


Figure 2.11: More and more points (with $x < 6$) in the infinite group $E(\mathbb{Q})$

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

$$[m] : E \rightarrow E, \quad P \mapsto [m]P,$$

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 $m = (m_9, \dots, m_0)_2 = (1, 0, 1, 0, 0, 0, 1, 1, 1, 1)$. 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).