We see that 3 is not a generator of the entire group. Rather, it generates a *subgroup* $\mathbb{G} = \{1, 3, 4, 5, 9\}$ of order 5. Now, let's see what happens with 10:

Powers of 10:	10^0	10^1	10^2	10^3	10^4	10^5	10^6	10^7	10^8	10^9
Values:	1	10	1	10	1	10	1	10	1	10

In this case we generate a subgroup of order 2.

For cryptographic purposes we want to work in a prime-order group. Since $11 = 2 \cdot 5 + 1$ we can apply Theorem 9.66 with $q = 5$ and $r = 2$, or with $q = 2$ and $r = 5$. In the first case, the theorem tells us that the *squares* of all the elements of $\mathbb{Z}_{11}^*$ should give a subgroup of order 5. This can be easily verified:

Element:	1	2	3	4	5	6	7	8	9	10
Square:	1	4	9	5	3	3	5	9	4	1

We have seen above that 3 is a generator of this subgroup. (In fact, since the subgroup has prime order, every element of the subgroup besides 1 is a generator of the subgroup.) Taking $q = 2$ and $r = 5$, Theorem 9.66 tells us that taking 5th powers will give a subgroup of order 2. One can check that this gives the order-2 subgroup generated by 10. ♢

Subgroups of finite fields. The discrete-logarithm problem is also believed to be hard in the multiplicative group of a finite field of large characteristic when the polynomial representation is used. (Appendix A.5 provides a brief background on finite fields.) Recall that for any prime p and integer $k \geq 1$ there is a (unique) field $\mathbb{F}_{p^k}$ of order p^k; the multiplicative group $\mathbb{F}_{p^k}^*$ of that field is a cyclic group of order $p^k - 1$ (cf. Theorem A.21). If q is a large prime factor of $p^k - 1$, then Theorem 9.66 shows that $\mathbb{F}_{p^k}^*$ has a cyclic subgroup of order q. (The only property of $\mathbb{Z}_p^*$ we used in the proof of that theorem was that $\mathbb{Z}_p^*$ is cyclic.) This offers another choice of prime-order groups in which the discrete-logarithm and Diffie–Hellman problems are believed to be hard. Our treatment of $\mathbb{Z}_p^*$ in this section corresponds to the special case $k = 1$. (Appropriate choice of parameters for cryptographic applications when $k > 1$ is outside the scope of this book.)

9.3.4 Elliptic Curves

The groups we have concentrated on thus far have all been based directly on modular arithmetic. Another class of groups important for cryptography is given by groups consisting of *points on elliptic curves*. Such groups are especially interesting from a cryptographic perspective since, in contrast to $\mathbb{Z}_p^*$ or the multiplicative group of a finite field, there are currently no known sub-exponential time algorithms for solving the discrete-logarithm problem in appropriately chosen elliptic-curve groups. (See Section 10.4 for further discussion.) For cryptosystems based on the discrete-logarithm or Diffie–Hellman

assumptions, this means that implementations based on elliptic-curve groups can be much more efficient—in terms of both computation and, especially, communication—than implementations based on prime-order subgroups of $\mathbb{Z}_p^*$ at any given level of security. In this section we provide a brief introduction to elliptic-curve cryptography. A deeper understanding of the issues discussed here requires more sophisticated mathematics than we are willing to assume on the part of the reader. Those interested in further exploring this topic are advised to consult the references at the end of this chapter.

Throughout this section, let $p \geq 5$ be a prime.[2] For our purposes, an *elliptic curve* is defined by a cubic equation (modulo p) in two variables x and y; the *points* on the curve are the solutions to the equation. For example, consider an equation E in the variables x and y of the form

$$y^2 = x^3 + Ax + B \bmod p, \tag{9.2}$$

where $A, B \in \mathbb{Z}_p$ satisfy $4A^3 + 27B^2 \neq 0 \bmod p$. (This condition ensures that the equation $x^3 + Ax + B = 0 \bmod p$ has no repeated roots.) Equation (9.2) is called the *Weierstrass representation* of an elliptic curve, and any elliptic curve can be written in this form by applying an invertible affine transformation to the variables x and y. Let $E(\mathbb{Z}_p)$ denote the set of pairs $(x, y) \in \mathbb{Z}_p \times \mathbb{Z}_p$ satisfying the above equation along with a special value $\mathcal{O}$ whose purpose we will discuss shortly; that is,

$$E(\mathbb{Z}_p) \stackrel{\text{def}}{=} \big\{(x, y) \mid x, y \in \mathbb{Z}_p \text{ and } y^2 = x^3 + Ax + B \bmod p\big\} \cup \{\mathcal{O}\}.$$

The elements $E(\mathbb{Z}_p)$ are called the *points* on the *elliptic curve* E defined by Equation (9.2), and $\mathcal{O}$ is called the *point at infinity*.

Example 9.69

An element $y \in \mathbb{Z}_p^*$ is a *quadratic residue modulo* p if there is an $x \in \mathbb{Z}_p^*$ such that $x^2 = y \bmod p$; in that case, we say x is a *square root of* y. If y is not a quadratic residue then we say it is a *quadratic non-residue*. For $p > 2$ prime, exactly half the elements in $\mathbb{Z}_p^*$ are quadratic residues, and every quadratic residue has exactly two square roots. (See Section 15.4.1.)

Let $f(x) \stackrel{\text{def}}{=} x^3 + 3x + 3$ and consider the curve $E : y^2 = f(x) \bmod 7$. Each value of x for which $f(x)$ is a quadratic residue modulo 7 yields two points on the curve; values x where $f(x)$ is not a quadratic residue have no corresponding point on the curve; values of x for which $f(x) = 0 \bmod 7$ give one point on the curve. This allows us to determine the points on the curve:

- $f(0) = 3 \bmod 7$, a quadratic non-residue modulo 7.

[2] The theory can be adapted to deal with $p \in \{2, 3\}$ but this introduces additional complications. Elliptic curves can, in fact, be defined over arbitrary *fields* (cf. Section A.5), and our discussion largely carries over to fields of characteristic not equal to 2 or 3.

- $f(1) = 0 \bmod 7$, so we obtain the point $(1, 0) \in E(\mathbb{Z}_7)$.
- $f(2) = 3 \bmod 7$, a quadratic non-residue modulo 7.
- $f(3) = 4 \bmod 7$, a quadratic residue modulo 7 with square roots 2 and 5. This yields the points $(3, 2), (3, 5) \in E(\mathbb{Z}_7)$.
- $f(4) = 2 \bmod 7$, a quadratic residue modulo 7 with square roots 3 and 4. This yields the points $(4, 3), (4, 4) \in E(\mathbb{Z}_7)$.
- $f(5) = 3 \bmod 7$, a quadratic non-residue modulo 7.
- $f(6) = 6 \bmod 7$, a quadratic non-residue modulo 7.

Including the point at infinity $\mathcal{O}$, there are 6 points in $E(\mathbb{Z}_7)$. $\diamondsuit$

A useful way to think about $E(\mathbb{Z}_p)$ is to look at the graph of Equation (9.2) over the reals (i.e., the equation $y^2 = x^3 + Ax + B$ without reduction modulo p) as in Figure 9.2. This figure does not correspond exactly to $E(\mathbb{Z}_p)$ because, for example, $E(\mathbb{Z}_p)$ has a finite number of points ($\mathbb{Z}_p$ is, after all, a finite set) while there are an infinite number of solutions to the same equation if we allow x and y to range over all real numbers. Nevertheless, the picture provides useful intuition. In such a figure, one can think of the "point at infinity" $\mathcal{O}$ as sitting at the top of the y-axis and lying on every vertical line.

It can be shown that every line intersecting $E(\mathbb{Z}_p)$ at two points must also intersect it at a third point, where (1) a point P is counted twice if the line is tangent to the curve at P, and (2) the point at infinity is also counted when the line is vertical. This fact is used to define a binary operation, called "addition" and denoted by $+$, on points of $E(\mathbb{Z}_p)$ in the following way:

- The point $\mathcal{O}$ is defined to be an (additive) identity; that is, for all $P \in E(\mathbb{Z}_p)$ we define $P + \mathcal{O} = \mathcal{O} + P = P$.
- For two points $P_1, P_2 \neq \mathcal{O}$ on E, we evaluate their sum $P_1 + P_2$ by drawing the line through P_1, P_2 (if $P_1 = P_2$ then draw the line tangent to the curve at P_1) and finding the third point of intersection P_3 of this line with $E(\mathbb{Z}_p)$; the third point of intersection may be $P_3 = \mathcal{O}$ if the line is vertical. If $P_3 = (x, y) \neq \mathcal{O}$ then we define $P_1 + P_2 \stackrel{\text{def}}{=} (x, -y)$. (Graphically, this corresponds to reflecting P_3 in the x-axis.) If $P_3 = \mathcal{O}$ then $P_1 + P_2 \stackrel{\text{def}}{=} \mathcal{O}$.

If $P = (x, y) \neq \mathcal{O}$ is a point of $E(\mathbb{Z}_p)$, then $-P \stackrel{\text{def}}{=} (x, -y)$ (which is clearly also a point of $E(\mathbb{Z}_p)$) is the unique inverse of P. Indeed, the line through (x, y) and $(x, -y)$ is vertical, and so the addition rule implies that $P + (-P) = \mathcal{O}$. (If $y = 0$ then $P = (x, y) = (x, -y) = -P$ but then the tangent line at P will be vertical and so $P + (-P) = \mathcal{O}$ here as well.) Of course, $-\mathcal{O} = \mathcal{O}$.

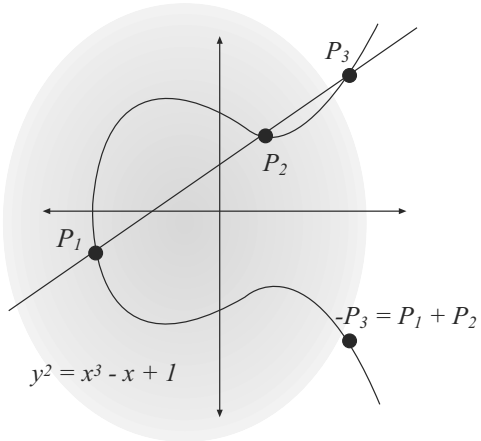

FIGURE 9.2: An elliptic curve over the reals.

It is straightforward, but tedious, to work out the addition law concretely for an elliptic curve in Weierstrass form. Let $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$ be two points in $E(\mathbb{Z}_p)$, with $P_1, P_2 \neq \mathcal{O}$ and E as in Equation (9.2). To keep matters simple, suppose $x_1 \neq x_2$ (dealing with the case $x_1 = x_2$ is even more tedious). The slope of the line through these points is

$$s \stackrel{\text{def}}{=} \left[\frac{y_2 - y_1}{x_2 - x_1} \bmod p \right];$$

our assumption that $x_1 \neq x_2$ means that the inverse of $(x_2 - x_1)$ modulo p exists. The line passing through P_1 and P_2 has the equation

$$y = s \cdot (x - x_1) + y_1 \bmod p\,. \tag{9.3}$$

To find the third point of intersection of this line with E, substitute the above into the equation for E to obtain

$$\Big(s \cdot (x - x_1) + y_1\Big)^2 = x^3 + Ax + B \bmod p.$$

The values of x that satisfy this equation are x_1, x_2, and

$$x_3 \stackrel{\text{def}}{=} [s^2 - x_1 - x_2 \bmod p\,].$$

The first two solutions correspond to the original points P_1 and P_2, while the third is the x-coordinate of the third point of intersection P_3. Plugging x_3 into Equation (9.3) we find that the y-coordinate corresponding to x_3 is $y_3 = [s \cdot (x_3 - x_1) + y_1 \bmod p\,]$. To obtain the desired answer $P_1 + P_2$, we flip the sign of the y-coordinate to obtain:

$$(x_1, y_1) + (x_2, y_2) = \big([s^2 - x_1 - x_2 \bmod p\,],\ [s \cdot (x_1 - x_3) - y_1 \bmod p\,]\big)\,.$$

We summarize and extend this in the following proposition.

PROPOSITION 9.70 *Let $p \geq 5$ be prime and let E be the elliptic curve given by $y^2 = x^3 + Ax + B \bmod p$ where $4A^3 + 27B^2 \neq 0 \bmod p$. Let $P_1, P_2 \neq \mathcal{O}$ be points on E, with $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$.*

1. *If $x_1 \neq x_2$, then $P_1 + P_2 = (x_3, y_3)$ with*

$$x_3 = [s^2 - x_1 - x_2 \bmod p] \quad \text{and} \quad y_3 = [s \cdot (x_1 - x_3) - y_1 \bmod p],$$

   where $s = \left[\frac{y_2 - y_1}{x_2 - x_1} \bmod p\right]$.

2. *If $x_1 = x_2$ but $y_1 \neq y_2$ then $P_1 = -P_2$ and so $P_1 + P_2 = \mathcal{O}$.*

3. *If $P_1 = P_2$ and $y_1 \neq 0$ then $P_1 + P_2 = 2P_1 = (x_3, y_3)$ with*

$$x_3 = [s^2 - x_1 - x_2 \bmod p] \quad \text{and} \quad y_3 = [s \cdot (x_1 - x_3) - y_1 \bmod p],$$

   where $s = \left[\frac{3x_1^2 + A}{2y_1} \bmod p\right]$.

4. *If $P_1 = P_2$ and $y_1 = 0$ then $P_1 + P_2 = 2P_1 = \mathcal{O}$.*

Somewhat amazingly, the set of points $E(\mathbb{Z}_p)$ under the addition rule defined above forms an abelian group, called the *elliptic-curve group of* $E(\mathbb{Z}_p)$. Commutativity follows from the way addition is defined, $\mathcal{O}$ acts as the identity, and we have already seen that each point in $E(\mathbb{Z}_p)$ has an inverse in $E(\mathbb{Z}_p)$. The difficult property to verify is associativity, which the disbelieving reader can check through tedious calculation. (A more illuminating proof that does not involve explicit calculation relies on algebraic geometry.)

Example 9.71
Consider the curve from Example 9.69. We show associativity for three specific points. Let $P_1 = (1, 0)$, $P_2 = P_3 = (4, 3)$. When computing $P_1 + P_2$ we get $s = [(3 - 0) \cdot (4 - 1)^{-1} \bmod 7] = 1$ and $[1^2 - 1 - 4 \bmod 7] = 3$. Thus,

$$Q \stackrel{\text{def}}{=} P_1 + P_2 = (3, \ [1 \cdot (1 - 3) - 0 \bmod 7]) = (3, 5);$$

note that this is indeed a point on $E(\mathbb{Z}_7)$. If we then compute $Q + P_3$ we get $s = [(3 - 5) \cdot (4 - 3)^{-1} \bmod 7] = 5$ and $[5^2 - 3 - 4 \bmod 7] = 4$. Thus,

$$(P_1 + P_2) + P_3 = Q + P_3 = (4, \ [5 \cdot (3 - 4) - 5 \bmod 7]) = (4, 4).$$

If we compute $P_2 + P_3 = 2P_2$ we obtain $s = [(3 \cdot 4^2 + 3) \cdot (2 \cdot 3)^{-1} \bmod 7] = 5$ and $[5^2 - 2 \cdot 4 \bmod 7] = 3$. Thus,

$$Q' \stackrel{\text{def}}{=} P_2 + P_3 = (3, \ [5 \cdot (4 - 3) - 3 \bmod 7]) = (3, 2).$$

If we then compute the value $P_1 + Q'$ we find $s = [2 \cdot (3-1)^{-1} \bmod 7] = 1$ and $[1^2 - 1 - 3 \bmod 7] = 4$. So

$$P_1 + (P_2 + P_3) = P_1 + Q' = (4,\ \ [1 \cdot (1-4) - 0 \bmod 7]) = (4, 4),$$

and $P_1 + (P_2 + P_3) = (P_1 + P_2) + P_3$. ♢

Recall that when a group is written additively, "exponentiation" corresponds to repeated addition. Thus, if we fix some point P in an elliptic-curve group, the discrete-logarithm problem becomes (informally) the problem of computing the integer x from xP, while the decisional Diffie–Hellman problem becomes (informally) the problem of distinguishing tuples of the form (aP, bP, abP) from those of the form (aP, bP, cP). These problems are believed to be hard in elliptic-curve groups (or subgroups thereof) of large prime order, subject to a few technical conditions we will mention below.

Montgomery representation. The Weierstrass representation is not the only way to define an elliptic curve, and other representations are often used for reasons of efficiency and/or implementation-level security (e.g., better resistance to side-channel attacks). The *Montgomery representation* involves equations of the form

$$By^2 = x^3 + Ax^2 + x \bmod p,$$

where $B \neq 0 \bmod p$ and $A \neq \pm 2 \bmod p$. Once again, given an equation E of the above form we let $E(\mathbb{Z}_p)$ denote the set of points (with coordinates in $\mathbb{Z}_p$) satisfying the equation plus the point at infinity $\mathcal{O}$; it is possible to define addition of these points in a way analogous to before. (Note that the addition law will *not* take the same form as in Proposition 9.70. Instead, addition is defined geometrically as before, and then the corresponding equations must be derived.) In contrast to the Weierstrass representation, not every curve can be expressed in Montgomery representation; in particular, the order of any elliptic-curve group written in Montgomery form is a multiple of 4.

(Twisted) Edwards representation. The *twisted Edwards representation* of an elliptic curve involves an equation E of the form

$$ax^2 + y^2 = 1 + dx^2y^2 \bmod p,$$

with $a, d \neq 0 \bmod p$ and $a \neq d \bmod p$; the special case where $a = 1$ is called the *Edwards representation*. The twisted Edwards representation can express the same set of elliptic curves as the Montgomery representation.

$E(\mathbb{Z}_p)$ again denotes the elliptic-curve group containing the points satisfying equation E; interestingly, here there is no need for a "special" point at infinity since one can show that the point $(0, 1)$ on the curve is the identity. A nice feature of the twisted Edwards representation is that when a is a quadratic

residue modulo p, but d is a quadratic non-residue, the addition law is simple: the sum of $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$ is

$$(x_3, y_3) = \left(\frac{x_1 y_2 + x_2 y_1}{1 + d x_1 x_2 y_1 y_2}, \frac{y_1 y_2 - a x_1 x_2}{1 - d x_1 x_2 y_1 y_2} \right).$$

Here addition is computed using a *single* equation, rather than having to consider various subcases as in Proposition 9.70. This can greatly simplify the process of writing code for elliptic-curve operations.

Choosing an elliptic-curve group. For cryptographic purposes, we need an elliptic-curve group of large order. Thus, the first question we must address is: how large are elliptic-curve groups? Consider the Weierstrass representation. (Recall that any elliptic curve can be expressed in that way.) As noted in Example 9.69, the equation $y^2 = f(x) \bmod p$ has two solutions whenever $f(x)$ is a quadratic residue, and one solution when $f(x) = 0$. Since half the elements in $\mathbb{Z}_p^*$ are quadratic residues, we thus heuristically expect to find $2 \cdot (p-1)/2 + 1 + 1 = p + 1$ points (including the point at infinity) on the curve. The *Hasse bound* says that this heuristic estimate is accurate, in the sense that every elliptic-curve group has "almost" this many points.

THEOREM 9.72 (Hasse bound) *Let p be prime, and let E be an elliptic curve over $\mathbb{Z}_p$. Then $p + 1 - 2\sqrt{p} \leq |E(\mathbb{Z}_p)| \leq p + 1 + 2\sqrt{p}$.*

In other words, we have $|E(\mathbb{Z}_p)| = p + 1 - t$ for $|t| \leq 2\sqrt{p}$. The value $t = p + 1 - |E(\mathbb{Z}_p)|$ is called the *trace* of the elliptic curve E. Efficient algorithms for computing the order (or, equivalently, the trace) of a given elliptic-curve group $E(\mathbb{Z}_p)$ are known, but are beyond the scope of this book.

The Hasse bound implies that it is always easy to *find* a point on a given elliptic curve $y^2 = f(x) \bmod p$: simply choose uniform $x \in \mathbb{Z}_p$, check whether $f(x)$ is 0 or a quadratic residue, and—if so—let y be a square root of $f(x)$. (Algorithms for deciding quadratic residuosity and computing square roots modulo a prime are discussed in Chapter 15.) Since points on the elliptic curve are plentiful, we will not have to try very many values of x before finding a point.

For cryptographic purposes, we want to work in an elliptic-curve (sub)group of prime order. If $|E(\mathbb{Z}_p)|$ is prime, we can simply work in the group $E(\mathbb{Z}_p)$. Otherwise, if $|E(\mathbb{Z}_p)|$ has a large prime factor then we can work in an appropriate subgroup of $E(\mathbb{Z}_p)$. Concretely, say $|E(\mathbb{Z}_p)| = rq$ with q prime and $r < q$ (so, in particular, $\gcd(r, q) = 1$); r is called the *cofactor*. Then it is possible to show that

$$\mathbb{G} \stackrel{\text{def}}{=} \{ rP \mid P \in E(\mathbb{Z}_p) \} \subset E(\mathbb{Z}_p)$$

is a subgroup of $E(\mathbb{Z}_p)$ of order q. (Note the parallel with Theorem 9.66, although here the larger group $E(\mathbb{Z}_p)$ may not be cyclic.)

Finally, we also want an elliptic-curve group in which the discrete-logarithm problem is as hard as possible, namely, for which the best-known algorithm for computing discrete logarithms in that group is an exponential-time "generic" algorithm. (See Section 10.2 for further discussion.) Several classes of elliptic curves are cryptographically weak and should be avoided. These include curves over $\mathbb{Z}_p$ whose order is equal to p (as discrete logarithms can be computed in polynomial time in that case), as well as curves whose order divides $p^k - 1$ for "small" k (since in that case the discrete-logarithm problem in $E(\mathbb{Z}_p)$ can be reduced to a discrete-logarithm problem in the field $\mathbb{F}_{p^k}$, which can in turn be solved by non-generic algorithms in sub-exponential time).

In practice, standardized curves recommended by NIST or other international standards organizations are used (see below); generating a curve of one's own for cryptographic purposes is not recommended.

Practical Considerations

We conclude this section with a brief discussion of some efficiency optimizations when using elliptic curves, and other practical aspects.

Point compression. A useful observation is that the number of bits needed to represent a point on an elliptic curve can be reduced almost by half. We illustrate the idea for curves using the Weierstrass representation. For any $x \in \mathbb{Z}_p$ there are at most two points on the curve with x as their x-coordinate: namely, $(x, \pm y)$ for some y. (It is possible that $y = 0$ in which case these are the same point.) Thus, we can specify any point $P = (x, y)$ on the curve by its x-coordinate and a bit b that distinguishes between the (at most) two possibilities for the value of its y-coordinate. One convenient way to do this is to set $b = 0$ if y is even and $b = 1$ if y is odd. Given x and b we can recover P by computing the two square roots y_1, y_2 of the equation $y^2 = f(x) \bmod p$; since $y_1 = -y_2 \bmod p$ and p is odd, either $y_1 = y_2 = 0$ or exactly one of y_1, y_2 will be even and the other will be odd.

Projective coordinates. Representing elliptic-curve points as we have been doing until now—in which a point P on an elliptic curve is described by a pair of elements (x, y)—is called using *affine coordinates*. There are alternate ways to represent points using *projective coordinates* that can offer efficiency improvements. While these alternate representations can be motivated mathematically, we treat them simply as useful computational aids. We continue to assume the Weierstrass representation for the elliptic curve.

Points in projective coordinates are represented using *three* elements of $\mathbb{Z}_p$. Specifically, a point $P = (x, y) \neq \mathcal{O}$ in affine coordinates is represented using (standard) projective coordinates by any tuple $(X, Y, Z) \in \mathbb{Z}_p^3$ for which $X/Z = x \bmod p$ and $Y/Z = y \bmod p$. (An interesting feature of using projective coordinates is that each point now has multiple representations.) The point at infinity $\mathcal{O}$ is represented by any tuple $(0, Y, 0)$ with $Y \neq 0$, and these are the only points (X, Y, Z) with $Z = 0$. We can easily translate between

coordinate systems: $(x, y) \neq \mathcal{O}$ in affine coordinates becomes $(x, y, 1)$ in projective coordinates, and (X, Y, Z) (with $Z \neq 0$) in projective coordinates is mapped to $([X/Z \bmod p], [Y/Z \bmod p])$ in affine coordinates.

The advantage of using projective coordinates is that we can add points without computing inverses modulo p. (Adding points in affine coordinates requires computing inverses; see Proposition 9.70. Although computing inverses modulo p can be done in polynomial time, it is more expensive than addition or multiplication modulo p.) This is done by exploiting the fact that points have multiple representations. To see this, let us work out the addition law for two points $P_1 = (X_1, Y_1, Z_1)$ and $P_2 = (X_2, Y_2, Z_2)$ with $P_1, P_2 \neq \mathcal{O}$ (so $Z_1, Z_2 \neq 0$) and $P_1 \neq \pm P_2$ (so $X_1/Z_1 \neq X_2/Z_2 \bmod p$). (If either P_1 or P_2 is equal to $\mathcal{O}$, addition is trivial. The case of $P_1 = \pm P_2$ can be handled as well, but we omit details here.) We can express P_1 and P_2 as $(X_1/Z_1, Y_1/Z_1)$ and $(X_2/Z_2, Y_2/Z_2)$, respectively, in affine coordinates, so (using Proposition 9.70)

$$\begin{aligned} P_3 \stackrel{\text{def}}{=} P_1 + P_2 = \; & \big(s^2 - X_1/Z_1 - X_2/Z_2, \\ & s \cdot (X_1/Z_1 - s^2 + X_1/Z_1 + X_2/Z_2) - Y_1/Z_1, \; 1\big), \end{aligned}$$

where

$$s = (Y_2/Z_2 - Y_1/Z_1)(X_2/Z_2 - X_1/Z_1)^{-1} = (Y_2 Z_1 - Y_1 Z_2)(X_2 Z_1 - X_1 Z_2)^{-1}$$

and all computations are done modulo p. Note we use projective coordinates (X_3, Y_3, Z_3) to represent P_3, setting $Z_3 = 1$ above. But using projective coordinates means we are not limited to $Z_3 = 1$. Multiplying each coordinate by $Z_1 Z_2 (X_2 Z_1 - X_1 Z_2)^3 \neq 0 \bmod p$, we find that P_3 can also be represented as

$$P_3 = \Big(vw, \;\; u(v^2 X_1 Z_2 - w) - v^3 Y_1 Z_2, \;\; Z_1 Z_2 v^3\Big) \tag{9.4}$$

where

$$\begin{aligned} u = Y_2 Z_1 - Y_1 Z_2, \qquad & v = X_2 Z_1 - X_1 Z_2, \\ w = u^2 Z_1 Z_2 - v^3 - 2v^2 X_1 Z_2. & \end{aligned} \tag{9.5}$$

The key observation is that the computations in Equations (9.4) and (9.5) can be carried out without having to perform any modular inversions.

Several other coordinate systems have also been developed, with the goal of minimizing the cost of elliptic-curve operations. Further details are beyond the scope of the book.

When points have multiple representations, some subtleties can arise. (Note that, until now, we have explicitly assumed that group elements have unique representations as bit-strings. That is no longer true when working in projective coordinates.) Specifically, a point expressed in projective coordinates may reveal information about how that point was computed, which may in turn leak some secret information. To address this, affine coordinates should

be used for transmitting and storing points, with projective coordinates used only as an intermediate representation during the course of a computation.

Popular elliptic curves. As noted earlier, in practice people typically do not generate their own elliptic curves, but instead use standardized curves that have been carefully selected to ensure both good security and efficient implementation. Some popular choices include:

- The *P-256 curve* (also known as secp256r1) is an elliptic curve over $\mathbb{Z}_p$ for the 256-bit prime $p = 2^{256} - 2^{224} + 2^{192} + 2^{96} - 1$. The prime was chosen to have this form because it allows for efficient implementation of arithmetic modulo p. The curve has the equation $y^2 = x^3 - 3x + B \bmod p$ where B is a specified constant; $A = -3$ was chosen to enable optimization of elliptic-curve operations. This curve has prime order (so cannot be represented using Montgomery or twisted Edwards form) that, by the Hasse bound, is of the same magnitude as p.

  P-384 (secp384r1) and P-521 (secp521r1) are analogous curves defined modulo 384- and 521-bit primes, respectively.

- *Curve25519* is an elliptic curve that can be represented in Montgomery form; it can also be represented in twisted Edwards form, where it is known as *Ed25519*. This curve is defined over $\mathbb{Z}_p$ for the 255-bit prime $p = 2^{255} - 19$, where again the prime was chosen to have this form because it allows for efficient implementation of arithmetic modulo p. This elliptic-curve group does not have prime order, but cryptographic operations can be carried out in a subgroup of large prime order.

- The *secp256k1* curve is a prime-order curve defined over $\mathbb{Z}_p$ where $p = 2^{256} - 2^{32} - 2^9 - 2^8 - 2^7 - 2^6 - 2^4 - 1$. This is a *Koblitz curve* with equation $y^2 = x^3 + 7 \bmod p$; a Koblitz curve has certain algebraic properties that allow for efficient implementation. This curve is most notable for being used by Bitcoin.

9.4 *Cryptographic Applications

We have spent a fair bit of time discussing number theory and group theory, and introducing computational hardness assumptions that are widely believed to hold. Applications of these assumptions will occupy us for the rest of the book, but we provide some brief examples here.