12

Elliptic Curve Cryptography

An elliptic curve is the set of solutions ("points") of an algebraic equation of the form $y^2 = x^3 + ax + b$, where a and b are numbers belonging to some field (for example, the real numbers, the rational numbers, or the integers mod p, or any finite field), and the variables x, y belong to the same field. Motivated by their graphs in the plane when the parameters are considered to be real numbers, a geometric procedure can be devised to define an "addition" operation on points on an elliptic curve, and this operation naturally extends to elliptic curves over any field. Endowed with this addition operation, elliptic curves provide a rich variety of new number systems. In the mid-1980s, number theorists Neal Koblitz (a professor at the University of Washington) and Victor Miller (then an IBM researcher) noticed that the discrete logarithm problems could be defined on these number systems and could be used as the basis of powerful public key cryptosystems that, for a given key size, tended to be much more robust and approximately 10 times more secure than all other known public key cryptosystems. This has important ramifications for efficient hardware implementations. Part of the reason for this security is the fact that, unlike for modular integers, there is no notion for "size" of points in modular elliptic curves. For example, the (elliptic curve) sum of two points with very small coordinates may have extremely large coordinates. At about the same time as Koblitz and Miller announced their results, Dutch mathematician Hendrik Lenstra discovered that prime factorization algorithms (for integers) could be created that are based on elliptic curve arithmetic, and these appeared to be more powerful than most integer-based factoring algorithms. These facts have transformed elliptic curves into one of the most extensively studied branches of cryptography. We begin by introducing elliptic curves over the real numbers and rigorously defining their addition operation by means of their graphs. We then discuss modular elliptic curves over $\mathbb{Z}_p$ and introduce the discrete logarithm problem for elliptic curves. This leads us to the development of natural extensions of the Diffie–Hellman key exchange and the ElGamal cryptosystems to the setting of modular elliptic curves. The chapter ends with an example of an elliptic curve-based factorization algorithm.

Elliptic Curves over the Real Numbers

Before studying elliptic curves over modular integers, it will be helpful to introduce them over the real numbers, as the latter possess many rich geometric properties that will provide some useful intuition and motivation.

Definition 12.1

Given a pair of real numbers $a, b \in \mathbb{R}$, the associated **elliptic curve E over the real numbers $\mathbb{R}$** is the set of all **points** represented by ordered pairs (x, y) of real numbers that solve the equation

$$y^2 = x^3 + ax + b \tag{12.1}$$

together with the **point at infinity**, which for brevity is denoted as ∞.* The **discriminant** of the elliptic curve is defined to be the following number:

$$\Delta \triangleq 4a^3 + 27b^2 \tag{12.2}$$

The elliptic curve is called **nonsingular** if its discriminant is nonzero: $\Delta \neq 0$; otherwise, it is called **singular**.

In order to better understand the soon-to-be-developed arithmetic of elliptic curves, we will take a moment to display the graphs of a few elliptic curves that illustrate the three general sorts of graphs that can arise.

Example 12.1

We consider elliptic curves over the real numbers where the parameter a of Equation 12.1 is taken to be -4: $y^2 = x^3 - 4x + b$.

(a) Determine the value of b for which the discriminant of the above elliptic curve is 0 (and thus the elliptic curve will be singular).

(b) Sketch planar graphs of the real-valued solutions of the above elliptic curve for the following parameter values: $b = 0, 2, 4, 6$.

(c) Sketch a planar graph of the real-valued solutions of the singular elliptic curve $y^2 = x^3 - 4x + b$, with the parameter b as determined in part (a).

Solution: Part (a): Setting the discriminant Δ of Equation 12.2 equal to zero, with $a = -4$, the equation is $4(-4)^3 + 27b^2 = 0 \Rightarrow b = \sqrt{4^4/27} \approx 3.0792\ldots$.

* Geometrically, the point at infinity should be thought of as lying infinitely far out in the plane, as we move away from zero in *any* direction.

Parts (b) and (c): We first make a few general observations about the planar graphs of the equation $y^2 = x^3 - 4x + b$. First, the x intercepts of the graph are determined by the root cubic equation $x^3 - 4x + b = 0$, and by the general facts of roots of polynomial equations, the roots will fall into one of three possibilities:*

(i) one real root (and two complex roots)
(ii) two distinct real roots (one of them a double root), or
(iii) three distinct real roots

Also, since $x^3 - 4x + b \to -\infty$ as $x \to -\infty$ and since we need to have $x^3 - 4x + b \geq 0$ for the equation $y^2 = x^3 - 4x + b$ to have a real-valued solution, it follows that there will be no solutions to $y^2 = x^3 - 4x + b$ when x is less than the smallest real root of $x^3 - 4x + b = 0$. Similarly, for each x larger than the largest real root of $x^3 - 4x + b = 0$, there will be exactly two values of y that satisfy $y^2 = x^3 - 4x + b$; these will be opposites and will get large (in absolute value) as x gets large. With this initial analysis, a computer graphing utility can then be used to produce the graphs of the four nonsingular elliptic curves for part (b) that are shown in Figure 12.1, and the singular elliptic curve of part (c) shown in Figure 12.2.

Comparing the graphs of the aforementioned two figures, it appears that when the parameter b is less than the critical value (≈ 3.08) that makes the discriminant zero, the elliptic curve consists of two pieces: a single loop (on the left) and a "boomerang"-shaped curve (on the right) that goes off to infinity.† As b approaches the critical value, these two pieces get closer to one another until they finally merge (Figure 12.2) when b reaches the critical value. Notice that at the meeting point of these two pieces (the intersection point of Figure 12.2), the curve does not have a well-defined tangent line.‡ When b is greater than the critical value, the intersection point vanishes and the elliptic curve consists of a single piece (see the labeled curves with $b = 4$ and $b = 6$ of Figure 12.1).

---

* From the *fundamental theorem of algebra,* any degree-n polynomial equation of the form $x^n + a_{n-1}x^{n-1} + \cdots + a_1x + a_0 = 0$, where the coefficients a_i are real numbers, will have exactly n roots, which may be real or complex numbers, some of which may be repeated. If the distinct roots are $r_1, r_2, \cdots, r_k$, and these are repeated with corresponding multiplicities $d_1, d_2, \cdots, d_k$, then we may write $x^n + a_{n-1}x^{n-1} + \cdots + a_1x + a_0 = (x - r_1)^{d_1}(x - r_2)^{d_2} \cdots (x - r_k)^{d_k}$. Also, any complex roots necessarily occur in conjugate pairs (with the same multiplicities) $r = \alpha \pm \beta i$, so the number of complex roots is even.

† It is clear from the graphs on page 454 that elliptic curves are not ellipses. The name comes from their appearance in so-called *elliptic integrals,* which are used to compute arc length of ellipses.

‡ Not having a tangent line at a point is the geometric definition for a planar curve to be *singular.* It can be shown that this property is equivalent to the discriminant being zero and to the polynomial equation $x^3 + ax + b = 0$ having at least one repeated root.

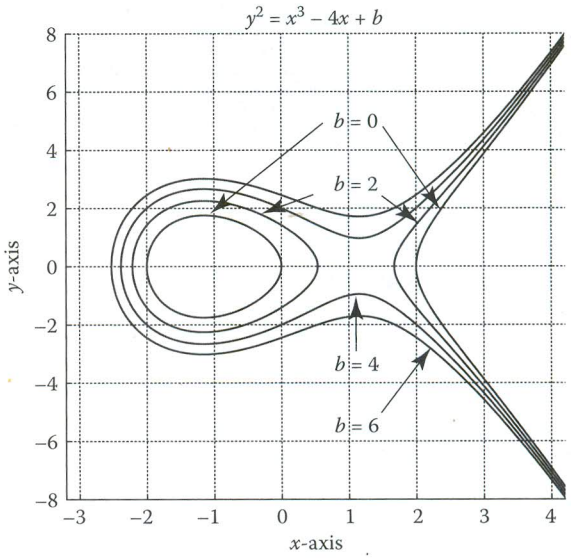

Figure 12.1 Graph of the four nonsingular elliptic curves $y^2 = x^3 - 4x + b$ with $b = 0, 2, 4, 6$. Note that for the values $b = 0, 2$ (that are less than the singular value of $b \approx 3.08$), the elliptic curve has two pieces, the loop on the left and the branch on the right, and three x intercepts, while for $b = 4, 6$ (that are greater than the singular value of $b \approx 3.08$), the elliptic curve has a single piece and a single x intercept.

The Addition Operation for Elliptic Curves

We assume that E is a nonsingular elliptic curve over the real numbers specified by the equation $y^2 = x^3 + ax + b$. We first give a geometric procedure showing how to add points on E, and then we will follow it with an algebraic formula.

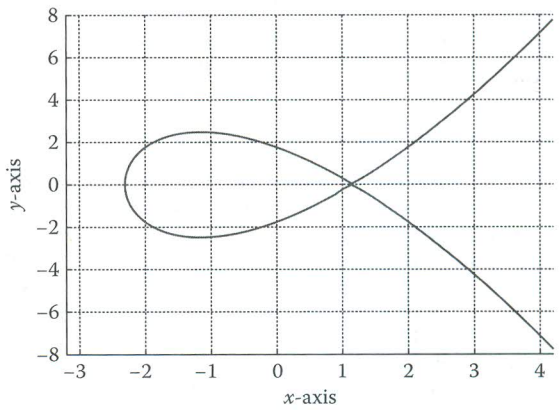

Figure 12.2 Graph of the singular elliptic curve $y^2 = x^3 - 4x + b$ with $b \approx 3.08$.

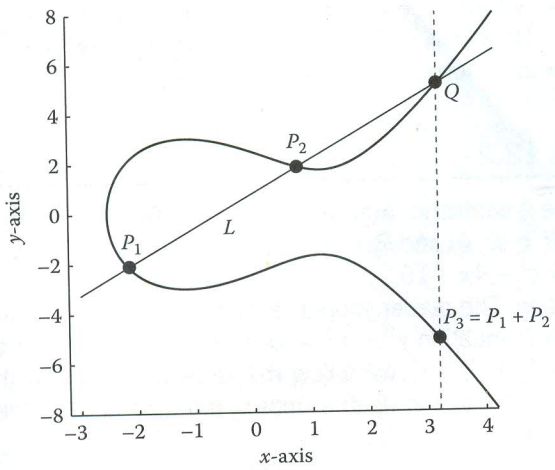

Figure 12.3 Illustration of the addition operation $P_3 = P_1 + P_2$ for two points on an elliptic curve E, in the case that $P_1 \neq P_2$, ∞; $P_2 \neq \infty$.

Algorithm 12.1: Addition of Points on Elliptic Curves over $\mathbb{R}$. Part I: Geometric Description of Algorithm via the Graph of the Elliptic Curve

Input: Two points, P_1, P_2, on a nonsingular elliptic curve E over the real numbers.

Output: A third point, $P_3 = P_1 + P_2$, on E (the **sum** of the first two points).

Case 1. $P_1 \neq P_2$, ∞; $P_2 \neq \infty$.

Step 1. Draw the line L through P_1, P_2. The line will intersect E in a (unique) third point Q; see Figure 12.3.

Step 2. Set the point $P_3 = P_1 + P_2$ to be the reflection of Q about the x axis; i.e., if $Q = (x, y)$, we set $P_3 = (x, -y)$. We make the convention that any vertical line passes through the point at infinity.

Case 2. $P_1 = P_2 \neq \infty$.
Use the procedure of Case 1, but with L taken to be the tangent line of the elliptic curve E at the point P_1. In this case, the line will intersect E in a (unique) second point Q.

Case 3. At least one of the two points $P_i = \infty$.

We use the convention that $P_1 + \infty = P_1$, $\infty + P_2 = P_2$.*

The corresponding algebraic formulation of the above algorithm can be derived using the relevant concepts about lines. Before presenting this

* We caution the reader (especially one who has studied calculus) that the point at infinity for elliptic curves behaves quite differently from the number infinity in the setting of real numbers, where we have $x + \infty = \infty$, for any real number x.

formulation, we give one specific example of an addition that will help to motivate it.* Of course, once we give the algebraic algorithm, there will be no need for these sorts of geometric derivations.

Example 12.2

Use the geometric Algorithm 12.1 to find the sum of the points $P_1 = (2, 4)$ and $P_2 = (4, 8)$ on the elliptic curve E defined by $y^2 = x^3 - 4x + 16$.

Solution: The reader should verify that both of the points P_1, P_2 satisfy the equation $y^2 = x^3 - 4x + 16$ and thus belong to E. To compute $P_3 = P_1 + P_2$, we follow the steps of Case 1 in the algorithm. We develop the algebra steps by referring to the entries of the points as follows: $P_1 = (2, 4) \triangleq (x_1, y_1)$ and $P_2 = (4,8) \triangleq (x_2, y_2)$.

Step 1. The line L through P_1, P_2 has slope $m = (y_2 - y_1)/(x_2 - x_1) = (8 - 4)/(4 - 2) = 2$. The equation of this line, $y = mx + b$, can be determined by substituting one of the points (we will use P_1) and the slope m into this equation to obtain $4 = 2 \cdot 2 + b \Rightarrow b = 0$. Thus, L has equation $y = 2x$. We then substitute this equation into that for E to find the unique third intersection point Q:

$$(2x)^2 = x^3 - 4x + 16 \Rightarrow x^3 - 4x^2 - 4x + 16 = 0$$

Since we already know two roots, $r_1 = 2$, $r_2 = 4$, we can easily determine the third root (that will be the x coordinate of Q) by using the general fact that the sum of the n roots of a degree-n polynomial equation having leading coefficient equal to 1 must always equal the negative of the x^{n-1} coefficient of the polynomial:† $r_1 + r_2 + r_3 = -(-4) \Rightarrow r_3 = 4 - r_1 - r_2 = 4 - 2 - 4 = -2$. This is the x-coordinate of Q; to obtain the corresponding y-coordinate, we substitute into the equation for L: $y = 2x = 2 \cdot (-2) = -4$. Thus $Q = (-2, -4)$.

Step 2. The point $P_3 = P_1 + P_2$ will be the reflection of Q about the x-axis: $P_3 = (-2, -(-4)) = (-2, 4)$.

We now present the general algebraic formulation of this addition algorithm.

* Our example illustrates Case 1 of the algorithm and requires only algebra. Case 2 would require finding the slope of the tangent line, and this can be accomplished by using some calculus (implicit differentiation of the elliptic curve equation).

† By the fundamental theorem of algebra (that was described in an earlier footnote), $x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0 = (x - r_1)(x - r_2)\cdots(x - r_n)$, where $r_1, r_2, \cdots, r_n$ are the roots (with multiple roots repeated). When we multiply out the right side, the only way to get monomials of degree $n - 1$ is to choose the term x in all but one of the factors, and to choose $-r_i$ from the remaining factor. It follows that when it is expanded and simplified, the term of degree $n - 1$ on the right will be $-(r_1 + r_2 + \cdots + r_n)x^{n-1}$, and if we equate the coefficient with the corresponding one of the original polynomial, we get that $r_1 + r_2 + \cdots + r_n = -a_{n-1}$.

Algorithm 12.1: Addition of Points on Elliptic Curves over $\mathbb{R}$. Part II: Algebraic Formulation

Inputs: Two points, $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$, on a nonsingular elliptic curve E over the real numbers,* which is defined by the equation $y^2 = x^3 + ax + b$.

Output: A third point, $P_3 = (x_3, y_3)$, on E that is the sum $P_1 + P_2$.

If either $P_1, P_2 = \infty$, output P_3 = the other point, and exit the algorithm.

Otherwise, we set (the slope of the line L): $m = \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1}, & \text{if } P_2 \neq P_2 \\ \dfrac{3x_1^2 + a}{2y_1}, & \text{if } P_2 = P_2 \end{cases}$

If m is undefined (i.e., if L is a vertical line), output $P_3 = \infty$, and exit the algorithm.

In all remaining cases, we set $\begin{cases} x_3 = m^2 - x_1 - x_2 \\ y_3 = m(x_1 - x_3) - y_1 \end{cases}$, and output $P_3 = (x_3, y_3)$.

Exercise for the Reader 12.1

Use the above algebraic algorithm to redo the computation of Example 12.2.

In earlier chapters, we experienced several sorts of "addition" and "multiplication" operations that were quite different from the usual ones with the same name. The next example shows that elliptic curve addition is different from vector addition even if all of the vectors belong to a given elliptic curve.

Example 12.3

Show that although (2, 4) and (4, 8) belong to the elliptic curve $y^2 = x^3 - 4x + 16$, we have $(2, 4) + (2, 4) \neq (4, 8)$.

Solution: Using the algebraic form of Algorithm 12.1 with $P_1, P_2 = (2, 4)$, we obtain $m = \dfrac{3 \cdot 2^2 - 4}{2 \cdot 4} = 1$, so we may compute $x_3 = m^2 - x_1 - x_2 = 1^2 - 2 - 2 = -3$, and then $y_3 = m(x_1 - x_3) - y_1 = 1(2 - (-3)) - 4 = 5 - 4 = 1$. Thus $P_3 = (-3, 1)$; that is, $(2, 4) + (2, 4) = (-3, 1) \neq (4, 8)$.

Although this addition operation may appear a bit contrived, it turns out to have some of the properties enjoyed by ordinary addition. For example,

* In this development, the point at infinity ∞ should be viewed sitting infinitely far up the y-axis yet above the whole x-axis so that every vertical line intersects it (a bit like the North Pole).

since we have (by the algorithm) $P + \infty = P$ for any point P on the curve, the point at infinity behaves like the additive identity zero for ordinary addition of real numbers. Also notice that from (either version of) Algorithm 12.1, it follows that the reflection of any point $P = (x, y)$ of E over the x-axis; that is, the point $(x, -y)$ satisfies $(x, y) + (x, -y) = \infty$—the additive identity for elliptic curve addition. (Notice that this still works when $y = 0$, in which case both points are the same.) The following proposition collects these two facts, along with two other useful properties of elliptic curve addition.

Proposition 12.1: Properties of Elliptic Curve Addition

Suppose that E is a nonsingular elliptic curve defined by $y^2 = x^3 + ax + b$. The addition operation of points of E defined by Algorithm 12.1 has the following properties, where P, Q, and R denote points of E:

(1) *Commutativity.* $P + Q = Q + P$
(2) *Associativity.* $(P + Q) + R = P + (Q + R)$
(3) *Additive Identity.* $P + \infty = P$
(4) *Additive Inverses.* There exists a point $-P$ in E, such that $P + (-P) = \infty$. Moreover, $-(x, y) = (x, -y)$, and $-\infty = \infty$.

We have already proven parts (3) and (4). The commutativity (1) follows easily from either the geometric or the algebraic form of Algorithm 12.1. For example, using the geometric approach, since the line L through P and Q is the same as the line through Q and P, it follows that the third point R (where L intersects the elliptic curve) is the same for both additions, thus, so is its reflection over the x axis, and hence, $P + Q = Q + P$. The proof of the associativity property (2) is much more involved than those that we just gave for the other three properties. Although it can be proved using the algebraic formulation of Algorithm 12.1, this method would be quite messy and require the separation into different cases—we do not recommend that any readers try this (even those who are very adept with algebraic manipulations). There are more elegant geometric approaches to proving the elliptic curve associative law, but such approaches are quite involved; the interested reader may find one such approach in the 11-page Section 2.4 in [Was-03].

Groups

Notice that the four properties of Proposition 12.1 are exactly the first four axioms for ring addition in Definition 10.2, with only a stylistic difference in that we are denoting the additive identity here by ∞ rather than 0. Any nonempty set G with a single binary operation on which these axioms are satisfied composes a very important type of number system in mathematics known as an *abelian group*.

Definition 12.2

An **abelian** (or commutative) **group** is a set G that is endowed with a single binary operation, which in this definition we denote by the symbol $\triangle$, and for which the following axioms hold, where a, b, c denote generic elements of G:

1. *Commutativity.* $a \triangle b = b \triangle a$.
2. *Associativity of Addition.* $(a \triangle b) \triangle c = a \triangle (b \triangle c)$.
3. *Identity.* There exists in G an **identity** element, denoted as e, that satisfies: $a \triangle e = a$.
4. *Inverses.* For each ring element a, there exists a corresponding **inverse**, denoted as a^{-1}, that satisfies $a \triangle a^{-1} = e$.

The adjective *abelian* refers to the commutativity Axiom (1). In more general treatments, nonabelian groups [that satisfy only Axioms (2) through (4) in the above definition] are considered. Since all groups under our consideration will be abelian groups, we often refer to these simply as "groups" (omitting the *abelian* adjective). Groups whose binary operations are denoted as additions (+) are usually called **additive groups**, while those whose binary operations are denoted as multiplications ($\cdot$) are called **multiplicative groups**. In an additive group, the identity is usually denoted as 0 (zero), and in a multiplicative group it is usually denoted as 1 (one). Also, in an additive group, the inverse of a group element a is usually denoted as $-a$. Note that for elliptic curves, although they are additive groups, the identity ∞ is not denoted as 0.

From Definition 10.2, it follows that any ring with just its addition operation forms an additive group. From Exercise for the Reader 10.1 (and some of the ring axioms of Definition 10.2), it follows that the set of invertible elements $R^{\times}$ in any ring forms a multiplicative group. From Proposition 2.1, it follows that any nonsingular elliptic curve over the real numbers forms an additive group.

Group theory is a vast and very well understood branch of mathematics, and the Diffie–Hellman key exchange as well as the ElGamal cryptosystem can be easily generalized to work with any finite group. Although elliptic curves of the real numbers will always be infinite groups, the soon-to-be-considered elliptic curves over modular integers (of prime modulus) provide us with a rich variety of finite groups on which we will build new cryptosystems.

Notation: For a positive integer n, the additive group analogue for powers $b^n = \underbrace{b \cdot b \cdots b}_{n \text{ times}}$, in multiplicative groups, is a repeated summation of the same element for which we use the following notation:

$$nP \text{ (or } n \cdot P) \triangleq \underbrace{P + P + \cdots + P}_{n \text{ times}} \tag{12.3}$$

This notation derives from a familiar arithmetic property; for example, $5x = x + x + x + x + x$. We will adhere to this additive notation in all of our work with elliptic curves.*

Exercise for the Reader 12.2

Using the notation (Equation 12.3), the result of Example 12.3 can be expressed as $2 \cdot (2, 4) = (-3, 1)$. Compute $3 \cdot (2, 4)$ and $4 \cdot (2, 4)$.

Elliptic Curves over $\mathbb{Z}_p$

Suppose that $p > 3$ is a prime. The definition of an elliptic curve E over $\mathbb{Z}_p$ is the same as the definition of elliptic curves over the real numbers, except that the points on the curve are pairs of mod p integers that satisfy the elliptic curve equation mod p.

Definition 12.3

Given a prime number $p > 3$ and a pair of real numbers $a, b \in \mathbb{R}$, the associated **elliptic curve E over $\mathbb{Z}_p$** (or simply the **elliptic curve mod p**) is the set of all ordered pairs (x, y) of mod p integers that solve the congruence

$$y^2 \equiv x^3 + ax + b \pmod{p} \tag{12.4}$$

together with the **point at infinity** ∞. The **discriminant** of the elliptic curve is defined to be the following number:

$$\Delta \equiv 4a^3 + 27b^2 \pmod{p} \tag{12.5}$$

The elliptic curve is called **nonsingular** if its discriminant nonzero (mod p): $\Delta \not\equiv 0$; otherwise, it is called **singular**.

Note in contrast to elliptic curves over the real numbers, any *modular elliptic curve* E over $\mathbb{Z}_p$ is a finite set with at most $p^2 + 1$ elements, but we soon will provide a more accurate result.

Example 12.4

Verify that the following modular elliptic curve is nonsingular, and then find all of its points: the elliptic curve E is defined by $y^2 \equiv x^3 - 4x + 16 \equiv x^3 + x + 1 \pmod{5}$.

Solution: Since $\Delta \equiv 4a^3 + 27b^2 \equiv 4 \cdot 1^3 + 27 \cdot 1^2 \equiv 31 \equiv 1 \not\equiv 0 \pmod{5}$, the elliptic curve is nonsingular. A brute-force approach to finding the points of an elliptic curve mod a prime p would be

* In this regard, from a cryptographic perspective, it would make more sense to use multiplicative notation rather than addition notation in the elliptic curve operation. But addition of points on elliptic curves has been in existence much longer than these cryptographic applications, so the addition notation is here to stay.

TABLE 12.1 Finite Points (x, y) on the Elliptic Curve $y^2 \equiv x^3 + x + 1 \pmod 5$

x	$x^3 + x + 1 \pmod 5$	$y \equiv \sqrt{x^3 + x + 1} \pmod 5$
0	1	1, 4
1	3	None
2	1	1, 4
3	1	1, 4
4	4	2, 3

to simply let x run through the integers mod p: $0, 1, = \cdots, p-1$, evaluate $x^3 - 4x + 16 \pmod p$ for each such x, and look for any square roots y of this number (mod p). The modular elliptic curve will be the set of all ordered pairs (x, y) that arise in this way together with the point at infinity. These computations are summarized in Table 12.1.

Thus, we can write $E = \{(0, 1), (0, 4), (2, 1), (2, 4), (3, 1), (3, 4), (4, 2), (4, 3), \infty\}$.

Exercise for the Reader 12.3

Verify that the following modular elliptic curve is nonsingular, and then find all of its points: the elliptic curve E is defined by $y^2 \equiv x^3 + x + 1 \pmod 7$.

In our search for square roots in the solution of the above example, we witness that there are either no square roots or exactly two. This is true in general whenever we are dealing with prime moduli, except for the exceptional case that 0 has only one square root (itself). This is summarized in the following proposition. After this, we recall a formula from Proposition 2.14 (proven in the exercises of Chapter 2) that gives a more efficient means of extracting square roots with a prime modulus p, in the case that $p \equiv 3 \pmod 4$.

Proposition 12.2: Uniqueness of Square Roots Modulo a Prime

If p is an odd prime number, and $a \not\equiv 0 \pmod p$ is an integer mod p, then the equation $x^2 \equiv a \pmod p$, that is, $x = \sqrt{a} \pmod p$, has either no solutions or exactly two solutions (mod p). Also, 0 has only one square root mod p: $\sqrt{0} \equiv 0 \pmod p$.

Proof: We first deal with the case when $a = 0$, which is easy but requires a separate treatment. If $x^2 \equiv 0 \pmod p$, then $p \mid x^2$, so it follows from Euclid's lemma (Proposition 2.7) that $p \mid x$, so that $x \equiv 0 \pmod p$.

For the remainder of the proof, we assume that $a \not\equiv 0 \pmod p$. Let g be a primitive root mod p; such primitive roots exist by Theorem 8.7. Since (by Proposition 8.6) the modular powers $g, g^2, \cdots, g^{p-1}$ make up all nonzero integers mod p, if x is any nonzero integer mod p, we can write $x \equiv g^j$ and $a \equiv g^\ell \pmod p$ for unique integers $j, \ell \in \{1, 2, \cdots, p-1\}$. Using these representations, the congruence $x^2 \equiv a \pmod p$ becomes

$g^{2j} \equiv g^{\ell} \pmod{p}$. By Proposition 8.5(c), this latter congruence is equivalent to $2j \equiv \ell \pmod{p-1}$. Since p is odd, $d = \gcd(2, p-1) = 2$, and by Algorithm 2.3 (and its proof), this latter congruence will have either no solutions (in the case $d \nmid \ell$), in which case a has no square root, or exactly two solutions (in the case $d \mid \ell$), in which case a has exactly two square roots. □

We next put forth the following formula from Proposition 2.15, that provides the two square roots of nonzero modular integer a, whenever they exist, in case of a prime modulus $p \equiv 3 \pmod 4$:

$$\sqrt{a} \equiv \pm a^{(p+1)/4} \pmod{p}, \text{ in case } a \not\equiv 0 \pmod{p} \text{ has square roots} \quad (12.6)$$

Equation 12.6 considerably speeds up the brute-force search of points on elliptic curves mod a prime p, provided that $p \equiv 3 \pmod 4$. For each value of x, we need compute only one modular power (with a moderately sized exponent) and check whether it is a square root, rather than squaring all of the $p-1$ nonzero integers mod p. We demonstrate this faster method in the Example 12.5.

Example 12.5

Verify that the following modular elliptic curve is nonsingular, and then find all of its points: the elliptic curve E is defined by $y^2 \equiv x^3 - 4x + 16 \equiv x^3 + 7x + 5 \pmod{11}$.

Solution: Since $\Delta \equiv 4a^3 + 27b^2 \equiv 4 \cdot 7^3 + 27 \cdot 5^2 \equiv 2047 \not\equiv 0 \pmod{11}$, the elliptic curve is nonsingular. We let x run through the integers mod 11: $0, 1, \cdots, 10$, evaluate $r \equiv x^3 + 7x + 5 \pmod{11}$ for each such x, and whenever the value of r is nonzero, we use Equation 12.6 to find any square roots of a. The modular elliptic curve will be the set of all ordered pairs (x, y) that arise in this way together with the point at infinity. The computations are summarized in Table 12.2.

From these computations, we may write $E = \{(0, 4), (0, 7), (2, 4), (2, 7), (3, 3), (3, 8), (4, 3), (4, 8), (5, 0), (7, 1), (7, 10), (8, 1), (8, 10), (9, 4), (9, 7), \infty\}$.

Exercise for the Reader 12.4

Verify that the following modular elliptic curve is nonsingular, and then use the method of the above example to find all of its points: the elliptic curve E is defined by $y^2 \equiv x^3 + x + 1 \pmod{11}$.

The Variety of Sizes of Modular Elliptic Curves

After working through the previous two examples, the following natural question arises: How many points are in a given elliptic curve modulo a prime $p > 3$? It turns out that there will be roughly p points; the following theorem contains a more precise result dating back to the 1930s.

TABLE 12.2 Determination of All Finite Points (x, y) on the Elliptic Curve $y^2 \equiv x^3 + 7x + 5 \pmod{11}$, Using Equation 12.6 to Compute Square Roots

x	$a \equiv x^3 + 7x + 5 \pmod{11}$	$a^3 \pmod{11}$	$(a^3)^2 \pmod{11}$	$y \equiv \pm a^3 \pmod{11}$?
0	5	4	5	$4, -4 \equiv 7$
1	2	8	9	No
2	5	4	5	$4, -4 \equiv 7$
3	9	3	9	$3, -3 \equiv 8$
4	9	3	9	$3, -3 \equiv 8$
5	0	–	–	0
6	10	10	1	No
7	1	1	1	$1, -1 \equiv 10$
8	1	1	1	$1, -1 \equiv 10$
9	5	4	5	$4, -4 \equiv 7$
10	8	6	3	No

Theorem 12.3: Hasse's Theorem

If E is an elliptic curve mod p, where $p > 3$ is a prime, then $|E|$ = the number of points in E satisfies the following inequalities:

$$p + 1 - 2\sqrt{p} \leq |E| \leq p + 1 + 2\sqrt{p} \tag{12.7}$$

A proof of this theorem can be found in Section 2.4 in [Was-03] or Section V.1 in [Sil-86]. It turns out that the range of possible values in Equation 12.7 are essentially all attainable for a given modulus p, for appropriate choices of the coefficients of the elliptic curve. This latter result is more recent; see [Wat-69].*

Theorem 12.4: Waterhouse's Theorem

If $p > 3$ is a prime and N is a positive integer that satisfies $p + 1 - 2\sqrt{p} \leq N \leq p + 1 + 2\sqrt{p}$, then there exists a nonsingular elliptic curve $E \pmod{p}$ with exactly N elements; that is, $|E| = N$.

The Addition Operation for Elliptic Curves over $\mathbb{Z}_p$

The addition operation for modular elliptic curves is defined using exactly the same formulas that were used in Algorithm 12.1, the only difference being that all divisions are performed mod p.†

* Waterhouse originally proved his theorem for N in the range $p + 1 - 2\sqrt{p} < N < p + 1 + 2\sqrt{p}$, but the result was extended in 1987 to hold also at the endpoints by Hans-Georg Rück; see [Rüc-87].

† Actually, the same definition works to define elliptic curves over any finite field $GF(p^n)$, but restricting attention to modular integer arithmetic will sufficiently illustrate the richness of elliptic curve cryptography, and the resulting cryptosystems are as secure as those that would result from using other finite fields (of comparable sizes).

Algorithm 12.2: Addition of Points on Elliptic Curves over $\mathbb{Z}_p$

Input: Two points, $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$, on a nonsingular elliptic curve E over $\mathbb{Z}_p$, which is defined by the congruence $y^2 \equiv x^3 + ax + b \pmod{p}$ where $p > 3$ is a prime number.

Output: A third point, $P_3 = (x_3, y_3)$, on E that is the **sum** $P_1 + P_2$.

If either $P_1, P_2 = \infty$, output $P_3 =$ the other point, and exit the algorithm. Otherwise, we set (the slope of the line L):

$$m \equiv \begin{cases} (y_2 - y_1) \cdot (x_2 - x_1)^{-1}, & \text{if } P_2 \neq P_2 \\ (3x_1^2 + a) \cdot (2y_1)^{-1}, & \text{if } P_2 = P_2 \end{cases} \pmod{p}.$$

If m is undefined (that is, inverse in the above formula does not exist), output $P_3 = \infty$, and exit the algorithm.

In all remaining cases, we set $\begin{cases} x_3 \equiv m^2 - x_1 - x_2 \\ y_3 \equiv m(x_1 - x_3) - y_1 \end{cases} \pmod{p}$ and output $P_3 = (x_3, y_3)$.

Example 12.6

(a) Use Algorithm 12.2 to find the sum of the points $P_1 = (2, 4)$ and $P_2 = (5, 0)$ on the modular elliptic curve E defined by $y^2 \equiv x^3 + 7x + 5 \pmod{11}$.

(b) Use Algorithm 12.2 to compute $2P$, where $P = (44, 29)$ is on the modular elliptic curve E defined by $y^2 \equiv x^3 + 9 \pmod{907}$.

Solution: Part (a): We use the notation of the algorithm $P_1 = (2, 4) \triangleq (x_1, y_1)$, $P_2 = (5, 0) \triangleq (x_2, y_2)$, and $P_3 \triangleq (x_3, y_3)$ (that will be determined). Since working mod 11 we have $(x_2 - x_1)^{-1} \equiv (5 - 2)^{-1} \equiv 3^{-1} \equiv 4$, we may compute $m = (y_2 - y_1) \cdot (x_2 - x_1)^{-1} \equiv (0 - 4) \cdot 4 \equiv -16 \equiv 6$. Next, we may compute the coordinates of the sum:

$$\begin{cases} x_3 = m^2 - x_1 - x_2 \equiv 6^2 - 2 - 5 \equiv 29 \equiv 7 \\ y_3 = m(x_1 - x_3) - y_1 \equiv 6(2 - 7) - 4 \equiv -30 - 4 \equiv -34 \equiv 10 \end{cases}$$

(mod 11) so that $P_3 \equiv (7, 10)$.

Part (b): We set $P = (44, 29) \triangleq (x_1, y_1) \triangleq (x_2, y_2)$ and $P_3 \triangleq (x_3, y_3)$ (that will be determined). We use the extended Euclidean algorithm (Algorithm 2.2) to compute the inverse of $2y_1 \equiv 2 \cdot 29 \equiv 58$ to be 735(mod 907). Thus $m \equiv (3x_1^2 + a) \cdot (2y_1)^{-1} \equiv (3 \cdot 44^2 + 0) \cdot 735 \equiv 538 \pmod{p}$, and hence

$$\begin{cases} x_3 = m^2 - x_1 - x_2 \equiv 6^2 - 2 - 5 \equiv 29 \equiv 7 \\ y_3 = m(x_1 - x_3) - y_1 \equiv 6(2 - 7) - 4 \equiv -30 - 4 \equiv -34 \equiv 10 \end{cases}$$

(mod 11) so that $P_3 \equiv (7, 10)$.

Exercise for the Reader 12.5

Perform the following modular elliptic curve operations (using Algorithm 12.2), where the ambient elliptic curve is defined by $y^2 \equiv x^3 + x + 1 \pmod{11}$:

(a) $(4, 5) + (0, 10)$
(b) $2(4, 5)$

Since there is no geometric analogue for addition of points on a modular curve, it is no longer obvious that the sum of two points (as defined by this algorithm) will really belong to the same elliptic curve. But this indeed turns out to be the case; moreover, all of the nice properties of Proposition 12.1 remain valid in this setting (as do their corresponding algebraic proofs).

Theorem 12.5

Any nonsingular modular elliptic curve E is a finite (abelian) group under its addition operation.

We next discuss a very useful general group theoretic concept that we experienced in Chapter 8 in the setting of the multiplicative groups $\mathbb{Z}_n^\times$.

Definition 12.4

If G is a finite abelian group and $a \in G$, then the **order** of a in G, denoted as $\text{ord}_G(a)$, is the smallest positive integer k such that:

- In additive group notation: $ka = 0$.
- In multiplicative group notation: $a^k = 1$.

Although it is not so obvious that such positive integers exist, the following general theorem provides assurance of this fact, as well as some precisions on the possible values of orders.

Theorem 12.6

If G is a finite group and $a \in G$, then the order of a in G must divide the number of elements in G; that is, $\text{ord}_G(a) \mid |G|$.

A proof of this theorem can be found in any book on abstract algebra; see, for example, [Her-96] or [Hun-96]. Note that in the case $G = \mathbb{Z}_n^\times$, this theorem becomes $\text{ord}_{\mathbb{Z}_n^\times}(a) \mid \phi(n)$, which we proved in Chapter 8 [except we denoted $\text{ord}_{\mathbb{Z}_n^\times}(a)$ simply as $\text{ord}_n(a)$].

Example 12.7

Compute $\text{ord}_E(0, 4)$ where E is the elliptic curve defined by $y^2 \equiv x^3 + 7x + 5 \pmod{11}$ of Example 12.5.

Solution: In Example 12.5, it was found that $|E| = 16$. Thus, by Theorem 12.6, the only possibilities for $\text{ord}_E(0, 4)$ are 2, 4,

8, or 16. Letting $P = (0, 4)$, we compute (using Algorithm 12.2): $2P = (3, 3)$, $4P = 2P + 2P = (9, 7)$, and $8P = 4P + 4P = (5, 0)$, and so it follows that $\text{ord}_E(0, 4) = 16$. (The reader may wish to check that $16P = \infty$.)

Exercise for the Reader 12.6

Compute the orders of the points (0, 4) and (2, 4) of the modular elliptic curve E of Example 12.4.

The point $P = (0, 4)$ on the elliptic curve of Example 12.6 had the maximum possible order; such a point is thus the analogue of primitive root in $\mathbb{Z}_p^\times$. The same proof that we gave in Chapter 8 (Proposition 8.6) shows that if P is any point on a nonsingular modular elliptic curve, and $k = \text{ord}_E(P)$, then the points P, $2P$, 3P, $\cdots$, $kP = \infty$ are distinct points of E, and all higher multiples of P will cycle back through these multiples. Thus, if P has maximum possible order $|E|$, then $\{P, 2P, 3P, \cdots, |E| P = \infty\} = E$. While these analogues of primitive roots do not always exist in elliptic curves, it is always feasible to find points of high order, and such points will be the basis for generalizing the ElGamal types of cryptosystems into the setting of elliptic curves.

The Discrete Logarithm Problem on Modular Elliptic Curves

We first recall the **discrete logarithm problem for $\mathbb{Z}_p^\times$**: Given a primitive root g (mod p), and given an integer $a \in \mathbb{Z}_p^\times$, the problem asks for the exponent j (that will be unique mod $p - 1$) for which $g^j \equiv a (\text{mod } p)$. Since modular elliptic curves do not always have analogues of primitive roots, we use the following more general formulation:

Definition 12.5

If $p > 3$ is a prime, E is a nonsingular elliptic curve mod p, and $P, Q \in E$, the **(elliptic curve) discrete logarithm problem** is to find a positive integer m for which $Q = m \cdot P$, if such an integer exists. In case a solution exists, any positive integer that works is called a **discrete logarithm** of Q in the **base** P.

Just as with the modular integer version, the discrete logarithm problem for elliptic curves is an intractable problem. In fact, the elliptic curve problem is more difficult than the mod p version due to the bizarre behavior of elliptic curve addition. One important difference is that when working with integers, there is a notion of size—for example, the number of modular digits. When two small modular integers are multiplied (say, two 12-digit integers mod a 1000-digit number), the result will be small. But with elliptic curve addition, there is no way to distinguish between small and large points. One can have, for example, two points with small

coordinates on an elliptic curve whose sum has very large coordinates. In short, modular integer multiplication is much easier to understand than elliptic curve point addition. All that is needed is to have an elliptic curve E and a point of high order; such parameters can always be constructed, as we discuss later.

Example 12.8

Letting E be the elliptic curve defined by $y^2 \equiv x^3 + 7x + 5 \pmod{11}$, solve the discrete logarithm problem $(7, 1) = m \cdot (0, 4)$.

Solution: We use the brute-force approach: we compute $2 \cdot (0, 4)$, $3 \cdot (0, 4)$, and so forth, until we obtain the element $(7, 1)$. From the previous example, we know that $\text{ord}_E((0, 4)) = 16$, so there will be a solution. In the worst case, we will need to compute up to $15 \cdot (0, 4)$ [since $16 \cdot (0, 4) = \infty$]. Using Algorithm 12.2, we obtain: $2 \cdot (0, 4) = (3, 3)$, $3 \cdot (0, 4) = 2 \cdot (0, 4) + (0, 4) = (2, 4)$, $4 \cdot (0, 4) = 3 \cdot (0, 4) + (0, 4) = (9, 7)$, $5 \cdot (0, 4) = 5 \cdot (0, 4) + (0, 4) = (7, 1)$, so we can stop; $m = 5$ is the desired discrete logarithm.

An Elliptic Curve Version of the Diffie–Hellman Key Exchange

We first review the original Diffie–Hellman key exchange algorithm (Algorithm 9.1) and then illustrate the corresponding modifications needed for the elliptic curve version using a table. To create a common secret key, Alice and Bob choose a large prime number p and a primitive root $g \pmod{p}$; these can be made public. Then Alice and Bob each choose (preferably randomly) a private integer, a or b, respectively, in the range $1 \le a, b < p - 1$. Alice and Bob each compute their modular power $A \equiv g^a$ or $B \equiv g^b \pmod{p}$ and send it to each other. Alice and Bob then compute the common key K now as $K \equiv B^a$ and $K \equiv A^b$, respectively. Table 12.3 illustrates the changes that are needed to turn this into an elliptic curve algorithm.

Before giving an example, we make a few comments about the elliptic curve version. As with the traditional version, the large primes can be randomly generated, as explained in Chapter 8. The elliptic curve E, $y^2 \equiv x^3 + ax + b \pmod{p}$, can then be randomly generated by generating mod p integers for its coefficients a and b.* The nonsingular condition $4a^3 + 27b^2 \not\equiv 0 \pmod{p}$ needs to be checked. If it fails, just generate a new set of parameters for E—but with a large prime p; chances are very small this condition will fail. Since elliptic curves need not have points of maximum possible order $|E|$, we aim only to generate a point G of high order. One quick way to generate a point G on E is to first randomly generate a mod p integer as a potential x coordinate x_1. Then we compute $x_1^3 + ax_1 + b \pmod{p}$ and check to see if it has a square root y_1. If we

* We have an inevitable collision of notation: the a and b in the elliptic curve version are coefficients of the elliptic curve, while in the mod p version, they represent Alice's and Bob's secret parameters.

TABLE 12.3 Corresponding Elements for Turning the Standard ($\mathbb{Z}_p$) Diffie–Hellman Key Exchange (Algorithm 9.1) into an Elliptic Curve Version

	$\mathbb{Z}_p$ Version	Elliptic Curve Version
Base parameters (public)	p = a large prime number g = a primitive root (mod p)	p = a large prime number E = an elliptic curve (mod p) G = a point of E of high order
Secret parameters:	(preferably random)	(preferably random)
Alice:	$1 \le a < p-1$	$\sqrt{p} < n_A < p - \sqrt{p}$
Bob:	$1 \le b < p-1$	$\sqrt{p} < n_B < p - \sqrt{p}$
Nonsecret parameters:		
Alice:	$A \equiv g^a \pmod{p}$	$A \equiv n_A G$
Bob:	$B \equiv g^a \pmod{p}$	$B \equiv n_B G$
Obtain common key:		(key is x coordinate of K)
Alice:	$K \equiv B^a \pmod{p}$	$K \equiv n_A B$
Bob:	$K \equiv A^b \pmod{p}$	$K \equiv n_B A$

restrict our prime p to satisfy $p \equiv 3 \pmod 4$, this square root check can be done quickly using Equation 12.6. Since any integer can have at most two square roots (mod p), it follows from Hasse's theorem that roughly half the time, there will be a square root y_1. We take $G = (x_1, y_1)$. We would like G to have high order so that the corresponding elliptic curve discrete logarithm problem will be hard to solve (this will make the system secure). Specific implementation details on how to find points of high order on an elliptic curve are a bit beyond the scope of this chapter, but we provide an outline of one such scheme. Without knowing the number of points on an elliptic curve, it is generally difficult to compute orders of specific points. If we know the number of points on the elliptic curve E, then the problem can be made simpler by Theorem 12.6. In particular, in case there is a prime number of points on E, then it follows that every point different from the point at infinity will have the maximum possible order equal to $|E|$. There are efficient algorithms for counting points on elliptic curves. The first such polynomial time algorithm is known as *Schoof's algorithm* and was discovered in 1985. It was later embellished by other mathematicians. So one possible scheme for generating elliptic curves with points of high order is to first fix a large prime p (of desired size) and continue the process of randomly generating elliptic curves mod p and checking their orders using a Schoof-type algorithm. (The number of points on a randomly generated elliptic curve mod p tend to be uniformly distributed in the range specified by Waterhouse's theorem.) These orders can be tested for primality using one of the tests of Chapter 8. Once an elliptic curve of prime order is found, any point different from the point at infinity will do. We refer the reader to [HaMeVa-04] and to [Was-03] for details. In the selection of secret parameters, the bounds on n_A, n_B were used to ensure they would be less than $|E|$ by Hasse's theorem, and randomly selected from a large range of numbers. The reason that Alice's and Bob's common keys are equal is because $n_A B \equiv n_A n_B G \equiv n_B n_A G \equiv n_B A \pmod p$. The resulting

key K will be a point of E, and thus a pair of modular integers. We used the x coordinate (modular) integer of this vector as the actual key, but any other deterministic choice could work—for example, the y coordinate or the sum (or product) of these two coordinates.

We next give a "small" parameter example to help better illustrate the concepts. The exercises in the computer implementation material at the end of the chapter will allow the reader to work with much larger examples.

Example 12.9

Let p be the prime number 11027.

(a) Randomly generate the coefficients a and b to determine an elliptic curve E defined by $y^2 \equiv x^3 + ax + b \pmod{p}$, and check that it is nonsingular. (Repeat if necessary.)
(b) Randomly generate a point G on E. (For this example, do not be concerned about the order G).
(c) Suppose that Alice chooses her secret parameter to be $n_A = 32$, and Bob takes his to be $n_B = 21$. (These are intentionally made smaller than what was recommended in Table 12.3 for ease of illustration.) Compute their resulting private Diffie–Hellman key created by the elliptic curve protocol of Table 12.3 in both ways.

Solution: Part (a): We randomly generated $a = 4601$ and $b = 548$. We compute $4a^3 + 27b^2 \equiv 9142 \not\equiv 0 \pmod{p}$ so that the elliptic curve $y^2 \equiv x^3 + 4601x + 548 \pmod{p}$ is nonsingular.

Part (b): We randomly generate an x coordinate as a mod p integer: $x_1 = 9954$. We then substitute this into the right side of the elliptic equation to obtain $r \triangleq x_1^3 + 4601x_1 + 548 \equiv 4618 \pmod{p}$. Since $p \equiv 3 \pmod 4$, we use Equation 12.6 to compute a possible square root of r: $y_1 \equiv r^{(p+1)/4} \equiv 8879 \pmod{p}$ (we used fast modular exponentiation). We then check if this is actually a square root of r by squaring it (mod p). Since this turns out to be the case, we have obtained a point $G = (x_1, y_1) = (9954, 8879) \in E$. By the way, $\text{ord}_E(G) = 1099$.

Part (c): We first compute $A \equiv n_A G \equiv 32 \cdot G$. The fastest approach is to keep doubling:

$$2G \equiv G + G \equiv (4023, 9690)$$

$$4G \equiv 2G + 2G \equiv (9395, 4193)$$

$$8G \equiv 4G + 4G \equiv (10151, 6531)$$

$$16G \equiv 8G + 8G \equiv (8344, 5801)$$

$$A \equiv 32G \equiv 16G + 16G \equiv (2652, 8449)$$

These computations can help us now to compute (although Alice should not share them with Bob—Why?):

$$B \equiv n_B G \equiv 21 \cdot G \equiv 16G + 4G + G \equiv (530, 2745) + G \equiv (202, 3553)$$

To compute the shared private key, Alice would compute $K \equiv n_A B \equiv 32 \cdot B$. We use the same approach as before:

$$2B \equiv B + B \equiv (943, 3104)$$
$$4B \equiv 2B + 2B \equiv (933, 2778)$$
$$8B \equiv 4B + 4B \equiv (2960, 45)$$
$$16B \equiv 8B + 8B \equiv (1009, 7768)$$
$$K \equiv 32B \equiv 16B + 16B \equiv (8814, 8359)$$

On Bob's end, he would compute $K \equiv n_B A \equiv 21 \cdot A$, and he could proceed as follows:

$$2A \equiv A + A \equiv (7962, 7796)$$
$$4A \equiv 2A + 2A \equiv (6913, 5918)$$
$$8A \equiv 4A + 4A \equiv (4308, 3378)$$
$$16A \equiv 8A + 8A \equiv (3892, 10524)$$
$$K \equiv 21A \equiv 16A + 4A + A \equiv (8979, 9865) + A \equiv (8814, 8359)$$

Both Alice and Bob have obtained the same vector K, from which they can read off their secret key as the first component: key = 8814.

Fast Integer Multiplication of Points on Modular Elliptic Curves

Notice that in performing the integer multiplications of points in the preceding example, we used an efficient strategy that was analogous to the fast modular exponentiation algorithm of Chapter 6. Indeed, this fast modular exponentiation algorithm (Algorithm 2.2) for computing large modular powers is easily adapted into the following algorithm (by simply converting the multiplicative notation to additive notation). Just as fast modular exponentiation was important in the efficient implementation of the mod integer versions of the Diffie–Hellman and ElGamal cryptosystems, this algorithm will serve as the computational basis for this and other elliptic curve cryptosystems.

Algorithm 12.3: Fast Integer Multiples of Points on Elliptic Curves

Input: A nonsingular modular elliptic curve E, a point $P \in E$, and a positive integer multiplier x.

Output: The point $Q = x \cdot P \in E$.

Step 1. Use Algorithm 6.1 to create the binary expansion of the exponent x: $x \sim [d_K \; d_{K-1} \; \cdots d_1 \; d_0]$ (base 2)

Step 2. <Repeatedly double the point P as we run through the binary digits d_k of x, including the result in the cumulative sum only when $d_k = 1$>

Set $Q = \infty$ <Initialize cumulative sum Q>
Set $D = P \pmod m$ <Initialize doubling>
FOR $k = 0$ TO K

IF $d_k = 1$
  Update $Q \rightarrow Q + D$
END <IF>
  Update $D \rightarrow 2D$ <Doubling need not be done when $k = K$>
END <k FOR>

Step 3. Output: Q.

Exercise for the Reader 12.7

Let p be the prime number 251.

(a) Let E be the elliptic curve defined by $y^2 \equiv x^3 + 196x + 98 \pmod p$. Verify that E is nonsingular. Then, noting that $p \equiv 3 \pmod 4$, use Equation 12.6 to see whether the x coordinate $x_1 = 28$ (which we randomly generated) gives rise to a point on E. If it does, let y_1 denote the resulting y coordinate with the positive sign in Equation 12.6, and let $G = (x_1, y_1) \in E$.

(b) Suppose that Alice chooses her secret parameter to be $n_A = 9$, and Bob takes his to be $n_B = 16$. (These are intentionally made smaller than what was recommended in Table 12.3 for ease of illustration.) Compute their resulting private Diffie–Hellman key created by the elliptic curve protocol of Table 12.3 in both ways.

Representing Plaintexts on Modular Elliptic Curves

The ideas that were used to extend the Diffie–Hellman key exchange to the elliptic curve setting allow us to do the same for the ElGamal cryptosystem of Chapter 9. But there is one issue that first needs to be ironed out: How do we represent plaintexts on an elliptic curve? This issue was completely straightforward in the traditional ElGamal setting since the plaintext space was $\mathbb{Z}_p$, and any plaintext can be represented using blocks of mod p integers. The points of a given elliptic curve mod p, on the other hand, have coordinates that are much less predictable and need not include any given specific mod p integer that appears in our plaintext. Moreover, it would be inefficient to compute all of the points on a given elliptic curve to help us determine a reasonable plaintext space. We will give a very efficient probabilistic algorithm that will be able to effectively represent plaintext through points on an elliptic curve. The algorithm succeeds with probability less than any specified value; for example, we could stipulate that we wanted the failure rate to be less than 10^{-20} percent, which is adequate for all practical purposes. The following probabilistic algorithm is due to Neal Koblitz; it is based on the fact that, given a nonsingular elliptic

curve E defined by $y^2 \equiv x^3 + ax + b$ modulo a prime $p > 3$, roughly half of the integers mod p will have square roots, so if we want to represent a given plaintext message* m as an x coordinate of a point on E, there is only about a 50 percent chance that this will be possible, that is, that $m^3 + am + b$ will have a square root (mod p). For a given positive integer K that satisfies $(m+1)K < p$, we can check through the list of integers $Km, Km+1, Km+2, \cdots, Km+(K-1)$ to see if any one can appear as an x coordinate of a point in E. If any such integer so appears, we can unambiguously use it to represent our plaintext m, because $m = \text{floor}([Km+i]/K)$, whenever $0 \le i < K$. Since the probability that any one of these integers will fail is 1/2, the probability that all such attempts will fail may be estimated as $(1/2)^K$. This can be made as small as we like by choosing K to be sufficiently large. Of course, larger values of K necessitate larger values for the prime p, but this fact will not be important in real applications since the primes used will typically have hundreds of digits. The algorithm assumes that $p \equiv 3 (\text{mod } 4)$, so that Equation 12.6 may be used to efficiently compute square roots.

Algorithm 12.4: Koblitz's Algorithm for Plaintext Representations on an Elliptic Curve mod p

Input: A prime $p > 3$ satisfying $p \equiv 3 (\text{mod } 4)$, an elliptic curve E (mod p) with equation $y^2 \equiv x^3 + ax + b$, a positive integer K (error tolerance parameter), and a nonnegative integer m (the plaintext) that satisfies $(m+1)K < p$.

Output: The point $P \in E$, whose x coordinate is of the form $Km+i$, with $0 \le i < K$, or a failure message.

Note: The estimated failure rate for this algorithm is $(1/2)^K$.

Step 1. Initialize $i = 0$.

Step 2. Set $x = Km + i$, $r \equiv x^3 + ax + b (\text{mod } p)$, and use fast modular exponentiation (Algorithm 6.5) to compute $y \equiv r^{(p+1)/4} (\text{mod } p)$.

Step 3. Check whether $y^2 \equiv r (\text{mod } p)$. If it is, output $P = (x, y)$, and exit the algorithm. If it is not, but $i < K - 1$, update $i \to i+1$, and return to Step 2. If it is not and $i = K - 1$, output failure message and exit algorithm.

Example 12.10

Let p be the prime number 307, and let E be the elliptic curve $y^2 \equiv x^3 + 22x + 153 (\text{mod } p)$.

* In previous chapters we usually represented plaintext messages with an uppercase P. But since in the context of elliptic curves, P typically is used to represent a generic point on an elliptic curve, we will use a lowercase m as the generic symbol for a plaintext message for the remainder of this chapter.

(a) Suppose that we wish to use Koblitz's Algorithm 12.4 to represent a positive integer plaintext m and that we would like the failure rate to be less than 1/1000. Find the smallest parameter K in the algorithm that will achieve this. For this value of K, what is the range of plaintext values for m for which the algorithm can be applied?

(b) Apply Algorithm 12.4 using the value of K found in part (a), and with $m = 22$.

Solution: Part (a): Since $2^{-9} \approx 0.0020$, and $2^{-10} \approx 0.00098$, we would use $K = 10$. The equation $(m+1)K < p$ thus implies $(m+1)10 < 307 \Rightarrow 10m < 297 \Rightarrow m < 29.7$, so the range of admissible values for m is $0 \le m \le 29$.

Part (b):

Step 1. Initialize $i = 0$.

Step 2. Set $x = Km + i = 10 \cdot 22 + 0 = 220$, $r \equiv 220^3 + 22 \cdot 220 + 153 \equiv 93 (\text{mod}\, p)$, and use fast modular exponentiation (Algorithm 6.5) to compute $y \equiv r^{(p+1)/4} \equiv 93^{77} \equiv 287 (\text{mod}\ p)$.

Step 3. Since $y^2 \equiv 93 \equiv r (\text{mod}\, p)$, we have found a plaintext representative on E in just one iteration; we output $P = (x, y) = (220, 287)$, and exit the algorithm.

Exercise for the Reader 12.8

Repeat the instructions tasks of Example 12.10, but with the desired failure rate to be less than 1/500, and with the prime changed to $p = 523$, but with all other parameters (including the elliptic curve equation) the same as in the example.

An Elliptic Curve Version of the ElGamal Cryptosystem

We first review the original ElGamal algorithm (Algorithm 9.4) and then illustrate the corresponding modifications needed for the elliptic curve version using a table. Alice needs to send Bob a message m. They publicly agree on the system parameters consisting of a large prime p and a primitive root g mod p. Alice and Bob each choose (preferably randomly) a private key integer, a or b, respectively, in the range $1 \le a, b < p - 1$. Alice and Bob each compute the modular power, $A \equiv g^a$ or $B \equiv g^b (\text{mod}\, p)$, which will be his or her public key. The ciphertext that Alice sends to Bob will be the ordered pair (A, C), where $C \equiv B^a m (\text{mod}\, p)$, which Alice computes using Bob's public key B. Bob may decrypt this message by computing $A^{p-1-b} C (\text{mod}\, p)$, which will be the original plaintext message m. Table 12.4 illustrates the changes that are needed to turn this into an elliptic curve algorithm.

TABLE 12.4 Corresponding Elements for Turning the Standard ($\mathbb{Z}_p$) ElGamal Cryptosystem (Algorithm 9.4) into an Elliptic Curve Version

	$\mathbb{Z}_p$ Version	Elliptic Curve Version
Base parameters (public)	p = a large prime number g = a primitive root (mod p)	p = a large prime number E = an elliptic curve (mod p) G = a point of E of high order
Secret parameters:	(preferably random)	(preferably random)
Alice:	$1 \le a < p-1$	$\sqrt{p} < n_A < p - \sqrt{p}$
Bob:	$1 \le b < p-1$	$\sqrt{p} < n_B < p - \sqrt{p}$
Nonsecret parameters:		
Alice:	$A \equiv g^a \pmod{p}$	$A \equiv n_A G$
Bob:	$B \equiv g^a \pmod{p}$	$B \equiv n_B G$
Encryption of plaintext message m from Alice to Bob:	Ciphertext: (A, C), where $C \equiv B^a m \pmod{p}$	First use Algorithm 12.4 to find a point $P \in E$ representing m. Ciphertext: (A, C), where $C = P + n_A B$
Decryption at Bob's end:	$m \equiv A^{p-1-b} C \pmod{p}$	$P = C - n_B A$ As explained in Algorithm 12.4, m can be easily obtained from P

Similar comments regarding the parameters that were made about the elliptic curve version of the Diffie–Hellman key exchange apply here. The following computation shows that the decryption scheme actually works:

$$C - n_B A = P + n_A B - n_B A = P + K - K = P$$

where we have used the fact that $n_A B$ and $n_B A$ both equal the Diffie-Hellman key K. The following is an example with small parameters to illustrate this concept.

Example 12.11

Let $p = 307$, let E be the (nonsingular) elliptic curve $y^2 \equiv x^3 + 22x + 153 \pmod{p}$, and let $P = (220, 287)$ be the plaintext representative point that was found in Example 12.10.

(a) Randomly generate a point G on E. (For this example, do not be concerned about the order G.)

(b) Suppose that Alice chooses her secret parameter to be $n_A = 32$, and Bob takes his to be $n_B = 54$. Go through the ElGamal encryption process that Alice would need to do to send Bob her message that is represented by the point P. What is the ciphertext?

(c) Go through the ElGamal decryption procedure that would need to be done at Bob's end to decrypt Alice's message.

Solution: Part (a): We randomly generate an x coordinate as a mod p integer: $x_1 = 167$. We then substitute this into the right side of the elliptic curve equation to obtain $r \triangleq x_1^3 + 22x_1 + 153 \equiv 109 \pmod{p}$. Since $p \equiv 3 \pmod 4$, we use Equation 12.6 to compute a possible square root of r: $y_1 \equiv r^{(p+1)/4} \equiv 118 \pmod{p}$ (we used fast modular exponentiation). We then check if this actually is a square root of r by squaring it (mod p). Since this turns out to be the case, we have obtained a point $G = (x_1, y_1) = (167, 118) \in E$.

Part (b): To encrypt, Alice needs Bob's public key, which he has been able to compute using Algorithm 12.3 to be $B = n_B G = 54G = (188, 55)$. Following the encryption procedure of Table 12.4, Alice multiplies Bob's public key number B by her secret parameter n_A (using Algorithm 12.3) to obtain $n_A B = (204, 265)$. Then (using Algorithm 12.2) she adds her plaintext point to this point to obtain $C = P + n_A B = (24, 136)$. Alice also needs to compute her public key, which could be computed using Algorithm 12.3 to be $A = n_A G = 32G = (30, 45)$. The resulting elliptic curve ElGamal ciphertext that Alice sends to Bob is the pair of points $(A, C) = ((30, 45), (24, 136))$.

Part (c): To decrypt, Bob would first compute (using Algorithm 12.3) $n_B A = (204, 265)$ (this is the Diffie–Hellman key). He would then subtract this from C (the second point of Alice's ciphertext), by using Algorithm 12.2 to add $-n_B A = (204, -265) \equiv (204, 42)$ to C: $C - n_B A = (204, -265) \equiv (220, 287) = P$, as was anticipated.

Exercise for the Reader 12.9

Repeat the instructions tasks of Example 12.11, but with the prime changed to $p = 523$, the plaintext representative point P to be what was found in the solution of Exercise for the Reader 12.8, but with all other parameters (including the elliptic curve equation) the same as in the example. For part (a), try this sequence of values of x_1 (in this order): 395, 402, 195 in the "random" generation of the point G. (These three mod p integers were randomly generated; at least one will work.)

This program of extending integer-based cryptosystems whose security is based on the discrete logarithm problem can be continued in the elliptic curve setting. For example, the ElGamal digital signature scheme (Algorithm 9.5) can also be adapted for elliptic curves. The interested reader may find further such examples in [Sti-06] and [TrWa-06].

A Factoring Algorithm Based on Elliptic Curves

Elliptic curves can be used in a very elegant and effective way in a factoring algorithm. This algorithm nicely complements and is more robust and effective than Pollard's $p-1$ algorithm (Algorithm 8.5). We first explain

how and why this ingenious algorithm of Hendrik Lenstra works, then we state it formally and give a specific example.

Suppose that we wish to factor a positive integer n that we suspect is composite. Although the algorithm works under general circumstances, to simplify this explanation we will assume that $n = pq$ is a product of distinct (large) primes, a particularly notorious sort of composite integer that is used in RSA. Although modular elliptic curves were defined only for prime moduli, we will temporarily "pretend" that n is prime, and we randomly generate a nonsingular elliptic curve E mod n. An easy way to do this would be to first randomly generate three mod n integers x, y, and a, and then determine the mod n integer b by the equation $y^2 \equiv x^3 + ax + b \pmod{n}$. When we are working with large values of p, q, and n (as is typical in difficult factorization problems), it will almost always be the case that such randomly generated curves are nonsingular, so we will not concern ourselves with such verifications. Thus, we have randomly generated an elliptic curve E mod n and a point $P = (x, y)$ on E. Actually, using the Chinese remainder theorem, E can be defined as a pair of elliptic curves mod p and mod q. We temporarily denote these latter two (true) elliptic curves as E_p and E_q. The algorithm involves "attempting" to compute the scalar multiple $B!P$ (using Algorithm 12.3), where B is a suitable positive integer. The reason why this might lead us to a factor of n is because the point P may have different orders in E_p and in E_q. If these orders are indeed different, and one of them does not have any large prime factors, then in the computation of multiples kP (in the chain of computations leading to $B!P$), it will probably happen that we will get $kP = \infty$ in one of E_p or E_q, and a finite point in the other. Now, although we are only doing our calculations mod n, this means that in the corresponding addition process (with Algorithm 12.2) we will have a nontrivial gcd in trying to compute the (slope) parameter m. This corresponds to m being infinite in one of E_p or E_q, and finite in the other, and in this case the denominator used to compute m will be a nontrivial factor (either p or q) of n. Of course, if the orders of P in both E_p and E_q each have large prime factors, then this method will not produce a factorization—but it can be repeated any number of times using newly generated elliptic curves. Herein lies the main (theoretical and practical) advantage over Pollard's $p-1$ algorithm. In order to work, the latter algorithm needed $n-1$ to be free of large prime factors. By Waterhouse's theorem (Theorem 12.4), the number of points on randomly generated elliptic curves can lie anywhere in the range $p+1\pm 2\sqrt{p}$, and it can furthermore be shown that these numbers tend to be uniformly distributed in this range when the curves are randomly generated. From these facts and from Theorem 12.6, it follows that this elliptic curve factorization algorithm requires only that n be near some integers that do not have any large prime factors. This is a much less stringent requirement than was needed in Pollard's $p-1$ algorithm. We formally state this algorithm, and then give a specific example.

Algorithm 12.5: Lenstra's Algorithm for Factorization Using Elliptic Curves

Input: An odd composite integer $n > 3$, a positive integer B, and a positive integer K (number of trials).

Output: Either a nontrivial factor of n or no output in case the algorithm does not find one.

Step 0. *Initialize Trial Counter.* Initialize $i = 1$.

Step 1. *Generate Elliptic Curve and Point.* Randomly generate three mod n integers x, y, and a, and then choose the mod n integer b by the equation for an elliptic curve E: $y^2 \equiv x^3 + ax + b \pmod{n}$. Set $P = (x, y)$.

Step 2. Use Algorithm 12.3 in conjunction with Algorithm 12.2 to compute $B!P \pmod{n}$ through the following sequence of steps: $1!P = P$, $2!P = 2(1!P), 3!P = 3(2!P), \cdots B!P = B([B-1]!P)$. At each intermediate application of Algorithm 12.2 in this process, keep track of $d = \gcd(\text{Den}(m), n)$, where Den($m$) is the denominator of m in the algorithm [so either $(x_2 - x_1)$ or $2y_1$]. If this ever turns out to give a nontrivial factor of n, i.e., if $1 < d < n$, then output d as a nontrivial factor and exit the algorithm.

Step 3. Update $i \rightarrow i+1$. If $i < K$, return to Step 2; otherwise, exit the algorithm.

We give a "small" example to illustrate this algorithm.

Example 12.12

Apply Lenstra's elliptic curve factoring algorithm with $B = 10$, and $K = 3$ to the integer $n = 345,283$.

Solution: With $i = 1$, we randomly generate the following integers (mod n): $x = 325604$, $y = 236075$, $a = 275656$. Solving $y^2 \equiv x^3 + ax + b$ for $b \pmod{n}$ produces $b = 290844$. We now proceed through the sequential computation of $10!P$, where $P = (x, y)$: We use Algorithm 12.3, but we stop if one of the denominators Den(m) is ever strictly between 1 and n. The computations $2!P$, $3!P$, $4!P$ all go through, but a nontrivial factor will be detected in the process of computing $5!P$ form $4!P = (201323, 330647)$. This factor 487 is a factor of n, and allows us to factor $n = 487 \cdot 709$.

Computing Note: Computer implementations of Lenstra's algorithm would be needed to effectively demonstrate its superiority over that of Pollard, or even a brute-force algorithm. Due to limitations of floating point arithmetic accuracy, it would be necessary to have symbolic capabilities on the computing platform in order to apply it to integers with more than, say, seven digits. Before applying any factoring algorithm to a given integer n, it is best to first apply one of the efficient primality tests of Chapter 8 (such as Miller–Rabin) to test whether n is prime (and hence already factored).

Chapter 12 Exercises

Note: In a few of the computational exercises involving elliptic curves over real numbers, some answers may involve noninteger real numbers.

In such cases, display all answers to four decimals, but use the default storage for subsequent machine computations.

1. Let E be the elliptic curve over the real numbers defined by $y^2 = x^3 + 8$, and let $P_1 = (1, 3)$ and $P_2 = (2, 4)$ (two points on E).
   (a) Compute $P_1 + P_2$ by using the algebraic formulation of Algorithm 12.1 (as in Example 12.3).
   (b) Compute $P_1 + P_2$ by using the geometric formulation of Algorithm 12.1 (as in Example 12.2).

2. Let E be the elliptic curve over the real numbers defined by $y^2 = x^3 + 2x + 3$, and let $P_1 = (-1, 0)$ and $P_2 = (3, 6)$ (two points on E).
   (a) Compute $P_1 + P_2$ by using the algebraic formulation of Algorithm 12.1 (as in Example 12.3).
   (b) Compute $P_1 + P_2$ by using the geometric formulation of Algorithm 12.1 (as in Example 12.2).

3. Let E be the elliptic curve over the real numbers defined by $y^2 = x^3 + 1$, and let $P = (0, 1)$ (a point on E).
   (a) Compute $2P$.
   (b) Compute $3P, 4P, 5P, 6P, 7P$.
   (c) Based on your calculations in parts (a) and (b), give a formula for nP, whenever n is a positive integer.

4. Let E be the elliptic curve over the real numbers defined by $y^2 = x^3 + 1$, and let $P = (2, 3)$ (a point on E).
   (a) Compute $2P$.
   (b) Compute $3P, 4P$.

5. For each prime modulus given below, let E be the elliptic curve defined by $y^2 = x^3 + 8 \pmod{p}$. Find all points on E:
   (a) $p = 5$
   (b) $p = 7$
   (c) $p = 11$

6. For each prime modulus given below, let E be the elliptic curve defined by $y^2 = x^3 + 3x + 3 \pmod{p}$. Find all points on E:
   (a) $p = 5$
   (b) $p = 7$
   (c) $p = 11$

7. Let E be the modular elliptic curve defined by $y^2 = x^3 + 6x \pmod{13}$. Show that E is nonsingular and then perform the following arithmetic operations on E:
   (a) $(4, 7) + (0, 0)$
   (b) $(4, 7) + (5, 5)$
   (c) $-(4, 7)$
   (d) $(4, 7) + (4, 7)$

8. Let E be the modular elliptic curve defined by $y^2 = x^3 + 3x \pmod{17}$. Show that E is nonsingular and then perform the following arithmetic operations on E:
   (a) $(4, 5) + (0, 0)$
   (b) $(6, 9) + (11, 15)$
   (c) $-(4, 12)$
   (d) $(8, 14) + (8, 14)$

9. Let E be the modular elliptic curve defined by $y^2 = x^3 + 3x + 3 \pmod{5}$.
   (a) Show that E is nonsingular.
   (b) Find all points of E (including the point at infinity).
   (c) Create an addition table for the points of E.

10. Let E be the modular elliptic curve defined by $y^2 = x^3 + 3 \pmod{5}$.
   (a) Show that E is nonsingular.
   (b) Find all points of E (including the point at infinity).
   (c) Create an addition table for the points of E.

11. Let E be the modular elliptic curve defined by $y^2 = x^3 + 2x + 1 \pmod{11}$. Show that E is nonsingular and then perform the following arithmetic operations on E.
   (a) $2(0, 1)$
   (b) $3(0, 1)$
   (c) $6(0, 1)$
   (d) $9(0, 1)$

12. Let E be the modular elliptic curve defined by $y^2 = x^3 + 2x + 1 \pmod{11}$. Show that E is nonsingular and then perform the following arithmetic operations on E.
   (a) $2(6, 8)$
   (b) $4(6, 8)$
   (c) $8(6, 8)$
   (d) $12(6, 8)$

13. Let E be the modular elliptic curve defined by $y^2 = x^3 + 2x + 1 \pmod{11}$. Compute the following orders:
   (a) $\text{ord}_E((0, 10))$
   (b) $\text{ord}_E((3, 1))$
   (c) $\text{ord}_E(\infty)$
   (d) $\text{ord}_E((8, 10))$

14. Let E be the modular elliptic curve defined by $y^2 = x^3 + 2x + 1 \pmod{11}$. Compute the following orders:
   (a) $\text{ord}_E((3, 1))$
   (b) $\text{ord}_E((1, 2))$
   (c) $\text{ord}_E(\infty)$
   (d) $\text{ord}_E((0, 10))$

15. Let E be the modular elliptic curve defined by $y^2 = x^3 + 6x + 3 (\text{mod } 7)$. Let $P = (2, 3)$, and then solve the following discrete logarithm problems on E:
    (a) $(2, 4) = m \cdot P$
    (b) $(4, 0) = m \cdot P$
    (c) $(5, 2) = m \cdot P$
    (d) $(5, 5) = m \cdot P$

16. Let E be the modular elliptic curve defined by $y^2 = x^3 + x + 6 (\text{mod } 7)$. Let $P = (1, 1)$, and then solve the following discrete logarithm problems on E:
    (a) $(2, 4) = m \cdot P$
    (b) $(4, 5) = m \cdot P$
    (c) $(6, 2) = m \cdot P$
    (d) $(1, 6) = m \cdot P$

17. Let E be the nonsingular modular elliptic curve defined by $y^2 = x^3 + 84x (\text{mod } 269)$. Compute the following scalar multiples of the point $P = (18, 9) \in E$:
    (a) $10 \cdot P$
    (b) $56 \cdot P$
    (c) $135 \cdot P$
    (d) $402 \cdot P$

18. Let E be the nonsingular modular elliptic curve defined by $y^2 = x^3 + 84 (\text{mod } 223)$. Compute the following scalar multiples of the point $P = (9, 12) \in E$:
    (a) $7 \cdot P$
    (b) $46 \cdot P$
    (c) $101 \cdot P$
    (d) $368 \cdot P$

19. Let p be the prime number 163, and let E be the elliptic curve defined by $y^2 \equiv x^3 + 22x + 153 (\text{mod } p)$.
    (a) Show that E is nonsingular. Then, noting that $p \equiv 3 (\text{mod } 4)$, use Equation 12.6 to see whether the x coordinate $x_1 = 28$ (which we randomly generated) gives rise to a point on E. If it does, y_1 denotes the resulting y coordinate with the positive sign in Equation 12.6, and let $G = (x_1, y_1) \in E$. If it does not, repeat with the values $x_1 = 94, 10$ until this construction produces such a point $G = (x_1, y_1) \in E$. (One of these will work.)
    (b) Suppose that Alice chooses her secret parameter to be $n_A = 19$, and Bob takes his to be $n_B = 41$. (These are intentionally made smaller than what was recommended in Table 12.3 for ease of illustration.) Compute their resulting private Diffie–Hellman key created by the elliptic curve protocol of Table 12.3 in both ways.
    (c) Repeat part (b) with $n_A = 192$ and $n_B = 94$.

20. Let p be the prime number 239, and let E be the elliptic curve defined by $y^2 \equiv x^3 + 39x + 58 \pmod{p}$.
    (a) Show that E is nonsingular. Then, noting that $p \equiv 3 \pmod{4}$, use Equation 12.6 to see whether the x coordinate $x_1 = 134$ (which we randomly generated) gives rise to a point on E. If it does, y_1 denotes the resulting y coordinate with the positive sign in Equation 12.6, and let $G = (x_1, y_1) \in E$. If it does not, repeat with the values $x_1 = 3, 96$, until this construction produces such a point $G = (x_1, y_1) \in E$. (One of these will work.)
    (b) Suppose that Alice chooses her secret parameter to be $n_A = 18$, and Bob takes his to be $n_B = 52$. (These are intentionally made smaller than what was recommended in Table 12.3 for ease of illustration.) Compute their resulting private Diffie–Hellman key created by the elliptic curve protocol of Table 12.3 in both ways.
    (c) Repeat part (b) with $n_A = 78$, and $n_B = 152$.

21. (a) Let E be the elliptic curve $y^2 \equiv x^3 + 39x + 58 \pmod{p}$, where $p = 431$. Use Koblitz's algorithm (Algorithm 12.4) to represent the plaintext message $m = 13$, using $K = 10$.
    (b) Repeat part (a) changing p to be 5431, m to 89, and using $K = 50$.
    (c) Repeat part (a) changing p to be 72379, m to 244, and using $K = 100$.

22. (a) Let E be the elliptic curve $y^2 \equiv x^3 + 62x + 9 \pmod{p}$, where $p = 311$. Use Koblitz's algorithm (Algorithm 12.4) to represent the plaintext message $m = 30$, using $K = 10$.
    (b) Repeat part (a) changing p to be 5431, m to 46, and using $K = 50$.
    (c) Repeat part (a) changing p to be 72379, m to 356, and using $K = 100$.

23. Let $p = 439$, let E be the (nonsingular) elliptic curve $y^2 \equiv x^3 + 6x + 167 \pmod{p}$, and let $P = (312, 65)$ be the plaintext representative point.
    (a) Noting that $p \equiv 3 \pmod{4}$, generate a point G on E by running through the x coordinates $x_1 = 38, 276, 61$, making use of Equation 12.6 (use the positive square root sign) until one is first found.
    (b) Suppose that Alice chooses her secret parameter to be $n_A = 24$, and Bob takes his to be $n_B = 71$. Go through the ElGamal encryption process that Alice would need to do to send Bob her message that is represented by the point P. What is the ciphertext?
    (c) Go through the ElGamal decryption procedure that would need to be done at Bob's end to decrypt Alice's message.
    (d) Repeat parts (a) and (b) using the following changes in secret parameters: $n_A = 89$, $n_B = 193$.

24. Let $p = 547$, let E be the (nonsingular) elliptic curve $y^2 \equiv x^3 + 6x + 167 \pmod{p}$, and let $P = (316, 521)$ be the plaintext representative point.
    (a) Noting that $p \equiv 3 \pmod 4$, generate a point G on E by running through the x coordinates $x_1 = 284, 341, 61$, making use of Equation 12.6 (use the positive square root sign) until one is first found.
    (b) Suppose that Alice chooses her secret parameter to be $n_A = 19$, and Bob takes his to be $n_B = 57$. Go through the ElGamal encryption process that Alice would need to do to send Bob her message that is represented by the point P. What is the ciphertext?
    (c) Go through the ElGamal decryption procedure that would need to be done at Bob's end to decrypt Alice's message.
    (d) Repeat parts (a) and (b) using the following changes in secret parameters: $n_A = 107$, $n_B = 150$.

25. In each part, a composite number n is specified. Apply Lenstra's elliptic curve factoring algorithm (Algorithm 12.5) using the parameters $B = 10$ and $K = 3$ with the aim of factoring n.
    (a) $n = 295{,}891$
    (b) $n = 1{,}544{,}927$
    (c) $n = 8{,}574{,}421$

    Note: It would be instructive for the reader to ignore the given fact that these integers are composite, and to apply one of the primality tests of Chapter 8 to ascertain these facts.

26. In each part, a composite number n is specified. Apply Lenstra's elliptic curve factoring algorithm (Algorithm 12.5) using the parameters $B = 10$ and $K = 3$ with the aim of factoring n.
    (a) $n = 288{,}619$
    (b) $n = 1{,}728{,}931$
    (c) $n = 11{,}064{,}199$

    Note: It would be instructive for the reader to ignore the given fact that these integers are composite, and to apply one of the primality tests of Chapter 8 to ascertain these facts.

27. Suppose that p is an odd prime number, $p > 3$ that satisfies $p \equiv 2 \pmod 3$.
    (a) Show that the function $f : \mathbb{Z}_p \to \mathbb{Z}_p$ defined by $f(x) \equiv x^3 \pmod p$ is a bijection.
    (b) Show that the number of points $|E|$ on the modular elliptic curve defined by $y^2 = x^3 + 1 \pmod p$ is exactly $p + 1$.

    Suggestion: For part (a), see Exercise 60 of Chapter 2. Use part (a) to prove part (b).

28. Without using Hasse's theorem, show that the number of points on $|E|$ on any modular elliptic curve E defined by $y^2 = x^3 + ax + b \pmod p$ is at most $2p + 1$.

29. Suppose that E is a nonsingular elliptic curve modulo a prime $p > 3$, and that the number of elements $|E|$ of E, is also a prime

number q. Show that for any $P \in E$, with $P \neq \infty$, we have $\text{ord}_E(P) = q$.

30. Suppose that E is a nonsingular elliptic curve over the real numbers.
    (a) Give a geometric condition that is equivalent to $\text{ord}_E(P) = 2$.
    (b) Give a geometric condition that is equivalent to $\text{ord}_E(P) = 3$.
    (c) Give a geometric condition that is equivalent to $\text{ord}_E(P) = 4$.

31. *Reflection Is Needed for Associativity.* Recall that in the geometric version of Algorithm 12.1, for adding two points $P_3 = P_1 + P_2$ on a nonsingular elliptic curve E over the real numbers, we first found the third point Q on the line L passing through P_1, P_2 and then reflected this point over the x axis to obtain the point P_3 (see Figure 12.3). Suppose that we define a new binary operation $\boxed{+}$ on E by using the same procedure, but skipping the reflection; i.e., we define $P_1 \boxed{+} P_2 \triangleq Q$.
    (a) Show this new binary operation $\boxed{+}$ is commutative.
    (b) Give an example to show that $\boxed{+}$ need not be associative. That is, find a specific nonsingular modular elliptic curve E and three points $P, Q, R \in E$ such that $(P \boxed{+} Q) \boxed{+} R \neq P \boxed{+} (Q \boxed{+} R)$.

Chapter 12 Computer Implementations and Exercises

Note: We reiterate a relevant principle that was stated in the computer implementation sections of Chapters 8 and 9: If your computing platform is a floating point arithmetic system, it may allow you only up to 15 or so significant digits of accuracy. Symbolic systems allow for much greater precision, being able to handle hundreds of significant digits. Some platforms allow users to choose if they wish to work in floating point or symbolic arithmetic. If you are working on such a dual-use platform, you may wish to create two separate programs for those who might work with large integers: an ordinary version and a symbolic version (perhaps attaching a `Sym` suffix to the names of those of the latter type). In case you do not have access to a symbolic system, some particular questions below may need to be skipped or modified so the numbers are of a manageable size.

1. *Adding Points on Nonsingular Elliptic Curves over the Real Numbers.*
    (a) Write a program with syntax `SumPoints = EllipticCurvePointAddition(P1, P2, a, b)` whose inputs `a`, `b` are integers that determine a nonsingular elliptic curve $y^2 = x^3 + ax + b$ over the real numbers (so that $4a^3 + 27b^2 \neq 0$), and the first two inputs `P1`, `P2` are length-2 vectors representing two points on this elliptic curve. The output, `SumPoints`, will be the length-2