Chapter 5

Elliptic Curves and Cryptography

The subject of elliptic curves encompasses a vast amount of mathematics.[1] Our aim in this section is to summarize just enough of the basic theory for cryptographic applications. For additional reading, there are a number of survey articles and books devoted to elliptic curve cryptography [14, 63, 72, 125], and many others that describe the number theoretic aspects of the theory of elliptic curves, including [25, 60, 68, 69, 123, 124, 127].

5.1 Elliptic curves

An *elliptic curve*[2] is the set of solutions to an equation of the form

$$Y^2 = X^3 + AX + B.$$

Equations of this type are called *Weierstrass equations* after the mathematician who studied them extensively during the 19^{th} century. Two examples of elliptic curves,

$$E_1 : Y^2 = X^3 - 3X + 3 \qquad \text{and} \qquad E_2 : Y^2 = X^3 - 6X + 5,$$

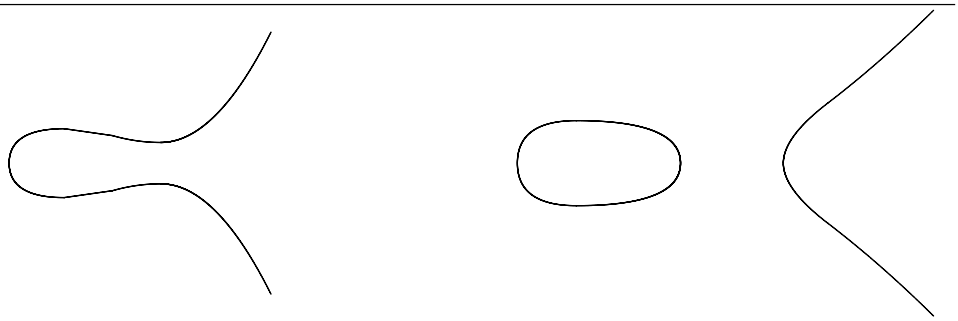
are illustrated in Figure 5.1.

An amazing feature of elliptic curves is that there is a natural way to take two points on an elliptic curve and "add" them to produce a third point. We

[1] Indeed, even before elliptic curves burst into cryptographic prominence, a well-known mathematician [68] opined that "it is possible to write endlessly on elliptic curves!"

[2] A word of warning. You may recall from high school geometry that an ellipse is a geometric object that looks like a squashed circle. Elliptic curves are *not* ellipses, and indeed, despite their somewhat unfortunate name, elliptic curves and ellipses have only the most tenuous connection with one another.

J. Hoffstein et al., *An Introduction to Mathematical Cryptography*,
DOI: 10.1007/978-0-387-77994-2_5,



$E_1 : Y^2 = X^3 - 3X + 3$ $E_2 : Y^2 = X^3 - 6X + 5$

Figure 5.1: Two examples of elliptic curves

put quotation marks around "add" because we are referring to an operation that combines two points in a manner analogous to addition in some respects (it is commutative and associative, and there is an identity), but very unlike addition in other ways. The most natural way to describe the "addition law" on elliptic curves is to use geometry.

Let P and Q be two points on an elliptic curve E, as illustrated in Figure 5.2. We start by drawing the line L through P and Q. This line L intersects E at three points, namely P, Q, and one other point R. We take that point R and reflect it across the x-axis (i.e., we multiply its Y-coordinate by -1) to get a new point R'. The point R' is called the "sum of P and Q," although as you can see, this process is nothing like ordinary addition. For now, we denote this strange addition law by the symbol $\oplus$. Thus we write[3]

$$P \oplus Q = R'.$$

Example 5.1. Let E be the elliptic curve

$$Y^2 = X^3 - 15X + 18. \tag{5.1}$$

The points $P = (7, 16)$ and $Q = (1, 2)$ are on the curve E. The line L connecting them is given by the equation[4]

$$L : Y = \frac{7}{3}X - \frac{1}{3}. \tag{5.2}$$

In order to find the points where E and L intersect, we substitute (5.2) into (5.1) and solve for X. Thus

[3]Not to be confused with the identical symbol $\oplus$ that we used to denote the XOR operation in a different context!

[4]Recall that the equation of the line through two points (x_1, y_1) and (x_2, y_2) is given by the point–slope formula $Y - y_1 = \lambda \cdot (X - x_1)$, where the slope λ is equal to $\frac{y_2 - y_1}{x_2 - x_1}$.

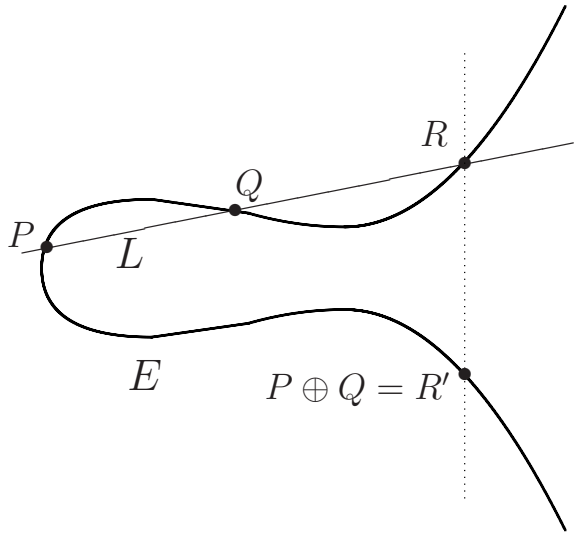


Figure 5.2: The addition law on an elliptic curve

$$\begin{aligned}\left(\frac{7}{3}X - \frac{1}{3}\right)^2 &= X^3 - 15X + 18,\\ \frac{49}{9}X^2 - \frac{14}{9}X + \frac{1}{9} &= X^3 - 15X + 18,\\ 0 &= X^3 - \frac{49}{9}X^2 - \frac{121}{9}X + \frac{161}{9}.\end{aligned}$$

We need to find the roots of this cubic polynomial. In general, finding the roots of a cubic is difficult. However, in this case we already know two of the roots, namely $X = 7$ and $X = 1$, since we know that P and Q are in the intersection $E \cap L$. It is then easy to find the other factor,

$$X^3 - \frac{49}{9}X^2 - \frac{121}{9}X + \frac{161}{9} = (X-7)\cdot(X-1)\cdot\left(X + \frac{23}{9}\right),$$

so the third point of intersection of L and E has X-coordinate equal to $-\frac{23}{9}$. Next we find the Y-coordinate by substituting $X = -\frac{23}{9}$ into equation (5.2). This gives $R = \left(-\frac{23}{9}, \frac{170}{27}\right)$. Finally, we reflect across the X-axis to obtain

$$P \oplus Q = \left(-\frac{23}{9}, -\frac{170}{27}\right).$$

There are a few subtleties to elliptic curve addition that need to be addressed. First, what happens if we want to add a point P to itself? Imagine what happens to the line L connecting P and Q if the point Q slides along the curve and gets closer and closer to P. In the limit, as Q approaches P, the line L becomes the tangent line to E at P. Thus in order to add P to

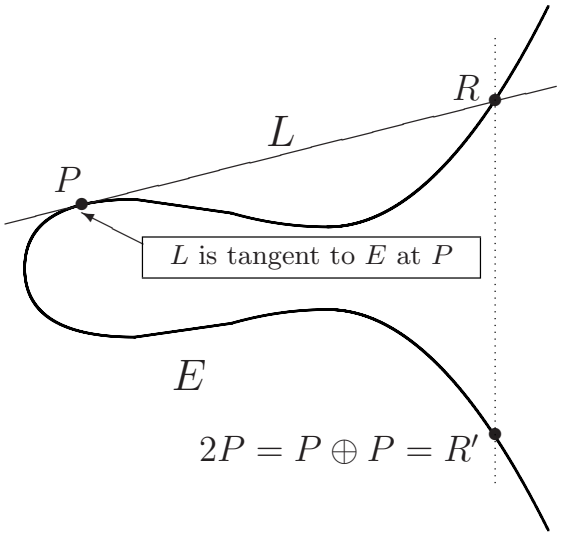


Figure 5.3: Adding a point P to itself

itself, we simply take L to be the tangent line to E at P, as illustrated in Figure 5.3. Then L intersects E at P and at one other point R, so we can proceed as before. In some sense, L still intersects E at three points, but P counts as two of them.

Example 5.2. Continuing with the curve E and point P from Example 5.1, we compute $P \oplus P$. The slope of E at P is computed by implicitly differentiating equation (5.1). Thus

$$2Y\frac{dY}{dX} = 3X^2 - 15, \qquad \text{so} \qquad \frac{dY}{dX} = \frac{3X^2 - 15}{2Y}.$$

Substituting the coordinates of $P = (7, 16)$ gives slope $\lambda = \frac{33}{8}$, so the tangent line to E at P is given by the equation

$$L : Y = \frac{33}{8}X - \frac{103}{8}. \tag{5.3}$$

Now we substitute (5.3) into the equation (5.1) for E, simplify, and factor:

$$\begin{aligned}\left(\frac{33}{8}X - \frac{103}{8}\right)^2 &= X^3 - 15X + 18,\\ X^3 - \frac{1089}{64}X^2 + \frac{2919}{32}X - \frac{9457}{64} &= 0,\\ (X-7)^2 \cdot \left(X - \frac{193}{64}\right) &= 0.\end{aligned}$$

Notice that the X-coordinate of P, which is $X = 7$, appears as a double root of the cubic polynomial, so it was easy for us to factor the cubic. Finally, we

substitute $X = \frac{193}{64}$ into the equation (5.3) for L to get $Y = -\frac{223}{512}$, and then we switch the sign on Y to get

$$P \oplus P = \left(\frac{193}{64}, \frac{223}{512}\right).$$

A second potential problem with our "addition law" arises if we try to add a point $P = (a, b)$ to its reflection about the X-axis $P' = (a, -b)$. The line L through P and P' is the vertical line $x = a$, and this line intersects E in only the two points P and P'. (See Figure 5.4.) There is no third point of intersection, so it appears that we are stuck! But there is a way out. The solution is to create an extra point $\mathcal{O}$ that lives "at infinity." More precisely, the point $\mathcal{O}$ does not exist in the XY-plane, but we pretend that it lies on every vertical line. We then set

$$P \oplus P' = \mathcal{O}.$$

We also need to figure out how to add $\mathcal{O}$ to an ordinary point $P = (a, b)$ on E. The line L connecting P to $\mathcal{O}$ is the vertical line through P, since $\mathcal{O}$ lies on vertical lines, and that vertical line intersects E at the points P, $\mathcal{O}$, and $P' = (a, -b)$. To add P to $\mathcal{O}$, we reflect P' across the X-axis, which gets us back to P. In other words, $P \oplus \mathcal{O} = P$, so $\mathcal{O}$ acts like zero for elliptic curve addition.

Example 5.3. Continuing with the curve E from Example 5.1, notice that the point $T = (3, 0)$ is on the curve E and that the tangent line to E at T is the vertical line $X = 3$. Thus if we add T to itself, we get $T \oplus T = \mathcal{O}$.

Definition. An *elliptic curve* E is the set of solutions to a Weierstrass equation

$$E : Y^2 = X^3 + AX + B,$$

together with an extra point $\mathcal{O}$, where the constants A and B must satisfy

$$4A^3 + 27B^2 \neq 0.$$

The *addition law on* E is defined as follows. Let P and Q be two points on E. Let L be the line connecting P and Q, or the tangent line to E at P if $P = Q$. Then the intersection of E and L consists of three points P, Q, and R, counted with appropriate multiplicities and with the understanding that $\mathcal{O}$ lies on every vertical line. Writing $R = (a, b)$, the sum of P and Q is defined to be the reflection $R' = (a, -b)$ of R across the X-axis. This sum is denoted by $P \oplus Q$, or simply by $P + Q$.

Further, if $P = (a, b)$, we denote the reflected point by $\ominus P = (a, -b)$, or simply by $-P$; and we define $P \ominus Q$ (or $P - Q$) to be $P \oplus (\ominus Q)$. Similarly, repeated addition is represented as multiplication of a point by an integer,

$$nP = \underbrace{P + P + P + \cdots + P}_{n \text{ copies}}.$$

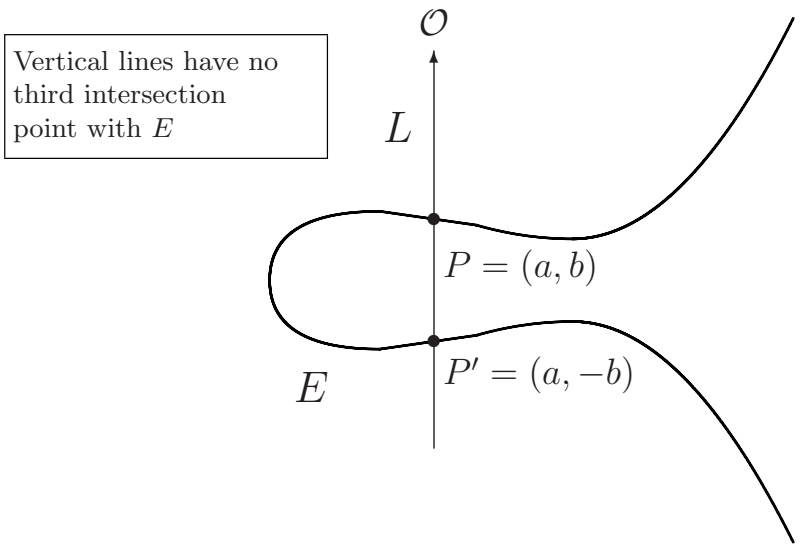


Figure 5.4: The vertical line L through $P = (a, b)$ and $P' = (a, -b)$

Remark 5.4. What is this extra condition $4A^3 + 27B^2 \neq 0$? The quantity $\Delta_E = 4A^3 + 27B^2$ is called the *discriminant of* E. The condition $\Delta_E \neq 0$ is equivalent to the condition that the cubic polynomial $X^3 + AX + B$ have no repeated roots, i.e., if we factor $X^3 + AX + B$ completely as

$$X^3 + AX + B = (X - e_1)(X - e_2)(X - e_3),$$

where e_1, e_2, e_3 are allowed to be complex numbers, then

$$4A^3 + 27B^2 \neq 0 \qquad \text{if and only if} \qquad e_1, e_2, e_3 \text{ are distinct.}$$

(See Exercise 5.3.) Curves with $\Delta_E = 0$ have singular points (see Exercise 5.4). The addition law does not work well on these curves. That is why we include the requirement that $\Delta_E \neq 0$ in our definition of an elliptic curve.

Theorem 5.5. *Let E be an elliptic curve. Then the addition law on E has the following properties:*

(a)	$P + \mathcal{O} = \mathcal{O} + P = P$	*for all* $P \in E$.	[Identity]
(b)	$P + (-P) = \mathcal{O}$	*for all* $P \in E$.	[Inverse]
(c)	$(P + Q) + R = P + (Q + R)$	*for all* $P, Q, R \in E$.	[Associative]
(d)	$P + Q = Q + P$	*for all* $P, Q \in E$.	[Commutative]

In other words, the addition law makes the points of E into an abelian group. (See Section 2.5 for a general discussion of groups and their axioms.)

Proof. As we explained earlier, the identity law (a) and inverse law (b) are true because $\mathcal{O}$ lies on all vertical lines. The commutative law (d) is easy to

verify, since the line that goes through P and Q is the same as the line that goes through Q and P, so the order of the points does not matter.

The remaining piece of Theorem 5.5 is the associative law (c). One might not think that this would be hard to prove, but if you draw a picture and start to put in all of the lines needed to verify (c), you will see that it is quite complicated. There are many ways to prove the associative law, but none of the proofs are easy. After we develop explicit formulas for the addition law on E (Theorem 5.6), you can use those formulas to check the associative law by a direct (but painful) calculation. More perspicacious, but less elementary, proofs may be found in [69, 123, 127] and other books on elliptic curves. □

Our next task is to find explicit formulas to enable us to easily add and subtract points on an elliptic curve. The derivation of these formulas uses elementary analytic geometry, a little bit of differential calculus to find a tangent line, and a certain amount of algebraic manipulation. We state the results in the form of an algorithm, and then briefly indicate the proof.

Theorem 5.6 (Elliptic Curve Addition Algorithm)**.** *Let*

$$E : Y^2 = X^3 + AX + B$$

be an elliptic curve and let P_1 and P_2 be points on E.
(a) *If $P_1 = \mathcal{O}$, then $P_1 + P_2 = P_2$.*
(b) *Otherwise, if $P_2 = \mathcal{O}$, then $P_1 + P_2 = P_1$.*
(c) *Otherwise, write $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$.*
(d) *If $x_1 = x_2$ and $y_1 = -y_2$, then $P_1 + P_2 = \mathcal{O}$.*
(e) *Otherwise, define λ by*

$$\lambda = \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1} & \text{if } P_1 \neq P_2, \\[2ex] \dfrac{3x_1^2 + A}{2y_1} & \text{if } P_1 = P_2, \end{cases}$$

and let

$$x_3 = \lambda^2 - x_1 - x_2 \qquad \text{and} \qquad y_3 = \lambda(x_1 - x_3) - y_1.$$

Then $P_1 + P_2 = (x_3, y_3)$.

Proof. Parts (a) and (b) are clear, and (d) is the case that the line through P_1 and P_2 is vertical, so $P_1 + P_2 = \mathcal{O}$. (Note that if $y_1 = y_2 = 0$, then the tangent line is vertical, so that case works, too.) For (e), we note that if $P_1 \neq P_2$, then λ is the slope of the line through P_1 and P_2, and if $P_1 = P_2$, then λ is the slope of the tangent line at $P_1 = P_2$. In either case the line L is given by the equation $Y = \lambda X + \nu$ with $\nu = y_1 - \lambda x_1$. Substituting the equation for L into the equation for E gives

$$(\lambda X + \nu)^2 = X^3 + AX + B,$$

so

$$X^3 - \lambda^2 X^2 + (A - 2\lambda\nu)X + (B - \nu^2) = 0.$$

We know that this cubic has x_1 and x_2 as two of its roots. If we call the third root x_3, then it factors as

$$X^3 - \lambda^2 X^2 + (A - 2\lambda\nu)X + (B - \nu^2) = (X - x_1)(X - x_2)(X - x_3).$$

Now multiply out the right-hand side and look at the coefficient of X^2 on each side. The coefficient of X^2 on the right-hand side is $-x_1 - x_2 - x_3$, which must equal $-\lambda^2$, the coefficient of X^2 on the left-hand side. This allows us to solve for $x_3 = \lambda^2 - x_1 - x_2$, and then the Y-coordinate of the third intersection point of E and L is given by $\lambda x_3 + \nu$. Finally, in order to get $P_1 + P_2$, we must reflect across the X-axis, which means replacing the Y-coordinate with its negative. □

5.2 Elliptic curves over finite fields

In the previous section we developed the theory of elliptic curves geometrically. For example, the sum of two distinct points P and Q on an elliptic curve E is defined by drawing the line L connecting P to Q and then finding the third point where L and E intersect, as illustrated in Figure 5.2. However, in order to apply the theory of elliptic curves to cryptography, we need to look at elliptic curves whose points have coordinates in a finite field $\mathbb{F}_p$. This is easy to do. We simply define an *elliptic curve over* $\mathbb{F}_p$ to be an equation of the form

$$E : Y^2 = X^3 + AX + B \qquad \text{with } A, B \in \mathbb{F}_p \text{ satisfying } 4A^3 + 27B^2 \neq 0,$$

and then we look at the points on E with coordinates in $\mathbb{F}_p$, which we denote by

$$E(\mathbb{F}_p) = \big\{(x, y) : x, y \in \mathbb{F}_p \text{ satisfy } y^2 = x^3 + Ax + B\big\} \cup \{\mathcal{O}\}.$$

Remark 5.7. For reasons that are explained later, we also require that $p \geq 3$. Elliptic curves over $\mathbb{F}_2$ are actually quite important in cryptography, but they are somewhat more complicated, so we delay our discussion of them until Section 5.7.

Example 5.8. Consider the elliptic curve

$$E : Y^2 = X^3 + 3X + 8 \quad \text{over the field } \mathbb{F}_{13}.$$

We can find the points of $E(\mathbb{F}_{13})$ by substituting in all possible values $X = 0, 1, 2, \ldots, 12$ and checking for which X values the quantity $X^3 + 3X + 8$ is a square modulo 13. For example, putting $X = 0$ gives 8, and 8 is not a square

modulo 13. Next we try $X = 1$, which gives $1+3+8 = 12$. It turns out that 12 is a square modulo 13; in fact, it has two square roots,

$$5^2 \equiv 12 \pmod{13} \qquad \text{and} \qquad 8^2 \equiv 12 \pmod{13}.$$

This gives two points $(1, 5)$ and $(1, 8)$ in $E(\mathbb{F}_{13})$. Continuing in this fashion, we end up with a complete list,

$$E(\mathbb{F}_{13}) = \{\mathcal{O}, (1,5), (1,8), (2,3), (2,10), (9,6), (9,7), (12,2), (12,11)\}.$$

Thus $E(\mathbb{F}_{13})$ consists of nine points.

Suppose now that P and Q are two points in $E(\mathbb{F}_p)$ and that we want to "add" the points P and Q. One possibility is to develop a theory of geometry using the field $\mathbb{F}_p$ instead of $\mathbb{R}$. Then we could mimic our earlier constructions to define $P + Q$. This can be done, and it leads to a fascinating field of mathematics called algebraic geometry. However, in the interests of brevity of exposition, we instead use the explicit formulas given in Theorem 5.6 to add points in $E(\mathbb{F}_p)$. But we note that if one wants to gain a deeper understanding of the theory of elliptic curves, then it is necessary to use some of the machinery and some of the formalism of algebraic geometry.

Let $P = (x_1, y_1)$ and $Q = (x_2, y_2)$ be points in $E(\mathbb{F}_p)$. We define the sum $P_1 + P_2$ to be the point (x_3, y_3) obtained by applying the elliptic curve addition algorithm (Theorem 5.6). Notice that in this algorithm, the only operations used are addition, subtraction, multiplication, and division involving the coefficients of E and the coordinates of P and Q. Since those coefficients and coordinates are in the field $\mathbb{F}_p$, we end up with a point (x_3, y_3) whose coordinates are in $\mathbb{F}_p$. Of course, it is not completely clear that (x_3, y_3) is a point in $E(\mathbb{F}_p)$.

Theorem 5.9. *Let E be an elliptic curve over $\mathbb{F}_p$ and let P and Q be points in $E(\mathbb{F}_p)$.*

(a) *The elliptic curve addition algorithm (Theorem 5.6) applied to P and Q yields a point in $E(\mathbb{F}_p)$. We denote this point by $P + Q$.*

(b) *This addition law on $E(\mathbb{F}_p)$ satisfies all of the properties listed in Theorem 5.5. In other words, this addition law makes $E(\mathbb{F}_p)$ into a finite group.*

Proof. The formulas in Theorem 5.6(e) are derived by substituting the equation of a line into the equation for E and solving for X, so the resulting point is automatically a point on E, i.e., it is a solution to the equation defining E. This shows why (a) is true, although when $P = Q$, a small additional argument is needed to indicate why the resulting cubic polynomial has a double root. For (b), the identity law follows from the addition algorithm steps (a) and (b), the inverse law is clear from the addition algorithm Step (d), and the commutative law is easy, since a brief examination of the addition algorithm shows that switching the two points leads to the same result. Unfortunately, the associative law is not so clear. It is possible to verify the associative law directly

using the addition algorithm formulas, although there are many special cases to consider. The alternative is to develop more of the general theory of elliptic curves, as is done in the references cited in the proof of Theorem 5.5. □

Example 5.10. We continue with the elliptic curve

$$E : Y^2 = X^3 + 3X + 8 \qquad \text{over } \mathbb{F}_{13}$$

from Example 5.8, and we use the addition algorithm (Theorem 5.6) to add the points $P = (9, 7)$ and $Q = (1, 8)$ in $E(\mathbb{F}_{13})$. Step (e) of that algorithm tells us to first compute

$$\lambda = \frac{y_2 - y_1}{x_2 - x_1} = \frac{8 - 7}{1 - 9} = \frac{1}{-8} = \frac{1}{5} = 8,$$

where recall that all computations[5] are being performed in the field $\mathbb{F}_{13}$, so $-8 = 5$ and $\frac{1}{5} = 5^{-1} = 8$. Next we compute

$$\nu = y_1 - \lambda x_1 = 7 - 8 \cdot 9 = -65 = 0.$$

Finally, the addition algorithm tells us to compute

$$\begin{aligned} x_3 &= \lambda^2 - x_1 - x_2 = 64 - 9 - 1 = 54 = 2, \\ y_3 &= -(\lambda x_3 + \nu) = -8 \cdot 2 = -16 = 10. \end{aligned}$$

This completes the computation of

$$P + Q = (1, 8) + (9, 7) = (2, 10) \qquad \text{in } E(\mathbb{F}_{13}).$$

Similarly, we can use the addition algorithm to add $P = (9, 7)$ to itself. Keeping in mind that all calculations are in $\mathbb{F}_{13}$, we find that

$$\lambda = \frac{3x_1^2 + A}{2y_1} = \frac{3 \cdot 9^2 + 3}{2 \cdot 7} = \frac{246}{14} = 1 \quad \text{and} \quad \nu = y_1 - \lambda x_1 = 7 - 1 \cdot 9 = 11.$$

Then

$$x_3 = \lambda^2 - x_1 - x_2 = 1 - 9 - 9 = 9 \quad \text{and} \quad y_3 = -(\lambda x_3 + \nu) = -1 \cdot 9 - 11 = 6,$$

so $P + P = (9, 7) + (9, 7) = (9, 6)$ in $E(\mathbb{F}_{13})$. In a similar fashion, we can compute the sum of every pair of points in $E(\mathbb{F}_{13})$. The results are listed in Table 5.1.

It is clear that the set of points $E(\mathbb{F}_p)$ is a finite set, since there are only finitely many possibilities for the X-and Y-coordinates. More precisely, there are p possibilities for X, and then for each X, the equation

[5]This is a good time to learn that $\frac{1}{5}$ is a *symbol* for a solution to the equation $5x = 1$. In order to assign a value to the symbol $\frac{1}{5}$, you must know where that value lives. In $\mathbb{Q}$, the value of $\frac{1}{5}$ is the usual number with which you are familiar, but in $\mathbb{F}_{13}$ the value of $\frac{1}{5}$ is 8, while in $\mathbb{F}_{11}$ the value of $\frac{1}{5}$ is 9. And in $\mathbb{F}_5$ the symbol $\frac{1}{5}$ is not assigned a value.

	$\mathcal{O}$	$(1,5)$	$(1,8)$	$(2,3)$	$(2,10)$	$(9,6)$	$(9,7)$	$(12,2)$	$(12,11)$
$\mathcal{O}$	$\mathcal{O}$	$(1,5)$	$(1,8)$	$(2,3)$	$(2,10)$	$(9,6)$	$(9,7)$	$(12,2)$	$(12,11)$
$(1,5)$	$(1,5)$	$(2,10)$	$\mathcal{O}$	$(1,8)$	$(9,7)$	$(2,3)$	$(12,2)$	$(12,11)$	$(9,6)$
$(1,8)$	$(1,8)$	$\mathcal{O}$	$(2,3)$	$(9,6)$	$(1,5)$	$(12,11)$	$(2,10)$	$(9,7)$	$(12,2)$
$(2,3)$	$(2,3)$	$(1,8)$	$(9,6)$	$(12,11)$	$\mathcal{O}$	$(12,2)$	$(1,5)$	$(2,10)$	$(9,7)$
$(2,10)$	$(2,10)$	$(9,7)$	$(1,5)$	$\mathcal{O}$	$(12,2)$	$(1,8)$	$(12,11)$	$(9,6)$	$(2,3)$
$(9,6)$	$(9,6)$	$(2,3)$	$(12,11)$	$(12,2)$	$(1,8)$	$(9,7)$	$\mathcal{O}$	$(1,5)$	$(2,10)$
$(9,7)$	$(9,7)$	$(12,2)$	$(2,10)$	$(1,5)$	$(12,11)$	$\mathcal{O}$	$(9,6)$	$(2,3)$	$(1,8)$
$(12,2)$	$(12,2)$	$(12,11)$	$(9,7)$	$(2,10)$	$(9,6)$	$(1,5)$	$(2,3)$	$(1,8)$	$\mathcal{O}$
$(12,11)$	$(12,11)$	$(9,6)$	$(12,2)$	$(9,7)$	$(2,3)$	$(2,10)$	$(1,8)$	$\mathcal{O}$	$(1,5)$

Table 5.1: Addition table for $E : Y^2 = X^3 + 3X + 8$ over $\mathbb{F}_{13}$

$$Y^2 = X^3 + AX + B$$

shows that there are at most two possibilities for Y. (See Exercise 1.34.) Adding in the extra point $\mathcal{O}$, this shows that $\#E(\mathbb{F}_p)$ has at most $2p+1$ points. However, this estimate is considerably larger than the true size.

When we plug in a value for X, there are three possibilities for the value of the quantity

$$X^3 + AX + B.$$

First, it may be a quadratic residue modulo p, in which case it has two square roots and we get two points in $E(\mathbb{F}_p)$. This happens about 50% of the time. Second, it may be a nonresidue modulo p, in which case we discard X. This also happens about 50% of the time. Third, it might equal 0, in which case we get one point in $E(\mathbb{F}_p)$, but this case happens very rarely.[6] Thus we might expect that the number of points in $E(\mathbb{F}_p)$ is approximately

$$\#E(\mathbb{F}_p) \approx 50\% \cdot 2 \cdot p + 1 = p + 1.$$

A famous theorem of Hasse, later vastly generalized by Weil and Deligne, says that this is true up to random fluctuations.

Theorem 5.11 (Hasse)**.** *Let E be an elliptic curve over $\mathbb{F}_p$. Then*

$$\#E(\mathbb{F}_p) = p + 1 - t_p \qquad \textit{with } t_p \textit{ satisfying} \quad |t_p| \le 2\sqrt{p}.$$

Definition. The quantity

$$t_p = p + 1 - \#E(\mathbb{F}_p)$$

appearing in Theorem 5.11 is called the *trace of Frobenius* for $E/\mathbb{F}_p$. We will not explain the somewhat technical reasons for this name, other than to say that t_p appears as the trace of a certain 2-by-2 matrix that acts as a linear transformation on a certain two-dimensional vector space associated to $E/\mathbb{F}_p$.

[6]The congruence $X^3 + AX + B \equiv 0 \pmod{p}$ has at most three solutions, and if p is large, the chance of randomly choosing one of them is very small.

Example 5.12. Let E be given by the equation

$$E : Y^2 = X^3 + 4X + 6.$$

We can think of E as an elliptic curve over $\mathbb{F}_p$ for different finite fields $\mathbb{F}_p$ and count the number of points in $E(\mathbb{F}_p)$. Table 5.2 lists the results for the first few primes, together with the value of t_p and, for comparison purposes, the value of $2\sqrt{p}$.

p	$\#E(\mathbb{F}_p)$	t_p	$2\sqrt{p}$
3	4	0	3.46
5	8	−2	4.47
7	11	−3	5.29
11	16	−4	6.63
13	14	0	7.21
17	15	3	8.25

Table 5.2: Number of points and trace of Frobenius for $E : Y^2 = X^3 + 4X + 6$

Remark 5.13. Hasse's theorem (Theorem 5.11) gives a bound for $\#E(\mathbb{F}_p)$, but it does not provide a method for calculating this quantity. In principle, one can substitute in each value for X and check the value of $X^3 + AX + B$ against a table of squares modulo p, but this takes time $\mathcal{O}(p)$, so is very inefficient. Schoof [110] found an algorithm to compute $\#E(\mathbb{F}_p)$ in time $O\big((\log p)^6\big)$, i.e., he found a polynomial-time algorithm. Schoof's algorithm was improved and made practical by Elkies and Atkin, so it is now known as the *SEA algorithm*. We will not describe SEA, which uses advanced techniques from the theory of elliptic curves, but see [111]. Also see Remark 5.32 in Section 5.7 for another counting algorithm due to Satoh that is designed for a different type of finite field.

5.3 The elliptic curve discrete logarithm problem (ECDLP)

In Chapter 2 we talked about the discrete logarithm problem (DLP) in the finite field $\mathbb{F}_p^*$. In order to create a cryptosystem based on the DLP for $\mathbb{F}_p^*$, Alice publishes two numbers g and h, and her secret is the exponent x that solves the congruence

$$h \equiv g^x \pmod{p}.$$

Let's consider how Alice can do something similar with an elliptic curve E over $\mathbb{F}_p$. If Alice views g and h as being elements of the group $\mathbb{F}_p^*$, then the discrete logarithm problem requires Alice's adversary Eve to find an x such that

$$h \equiv \underbrace{g \cdot g \cdot g \cdots g}_{x \text{ multiplications}} \pmod{p}.$$

In other words, Eve needs to determine how many times g must be multiplied by itself in order to get to h.

With this formulation, it is clear that Alice can do the same thing with the group of points $E(\mathbb{F}_p)$ of an elliptic curve E over a finite field $\mathbb{F}_p$. She chooses and publishes two points P and Q in $E(\mathbb{F}_p)$, and her secret is an integer n that makes

$$Q = \underbrace{P + P + P + \cdots + P}_{n \text{ additions on } E} = nP.$$

Then Eve needs to find out how many times P must be added to itself in order to get Q. Keep in mind that although the "addition law" on an elliptic curve is conventionally written with a plus sign, addition on E is actually a very complicated operation, so this elliptic analogue of the discrete logarithm problem may be quite difficult to solve.

Definition. Let E be an elliptic curve over the finite field $\mathbb{F}_p$ and let P and Q be points in $E(\mathbb{F}_p)$. The *Elliptic Curve Discrete Logarithm Problem* (ECDLP) is the problem of finding an integer n such that $Q = nP$. By analogy with the discrete logarithm problem for $\mathbb{F}_p^*$, we denote this integer n by

$$n = \log_P(Q)$$

and we call n the *elliptic discrete logarithm of Q with respect to P.*

Remark 5.14. Our definition of $\log_P(Q)$ is not quite precise. The first difficulty is that there may be points $P, Q \in E(\mathbb{F}_p)$ such that Q is not a multiple of P. In this case, $\log_P(Q)$ is not defined. However, for cryptographic purposes, Alice starts out with a public point P and a private integer n and she computes and publishes the value of $Q = nP$. So in practical applications, $\log_P(Q)$ exists and its value is Alice's secret.

The second difficulty is that if there is one value of n satisfying $Q = nP$, then there are many such values. To see this, we first note that there exists a positive integer s such that $sP = \mathcal{O}$. We recall the easy proof of this fact (cf. Proposition 2.13). Since $E(\mathbb{F}_p)$ is finite, the points in the list $P, 2P, 3P, 4P, \ldots$ cannot all be distinct. Hence there are integers $k > j$ such that $kP = jP$, and we can take $s = k - j$. The smallest such $s \geq 1$ is called the *order of P*. (Proposition 2.14 tells us that the order of P divides $\#E(\mathbb{F}_p)$.) Thus if s is the order of P and if n_0 is any integer such that $Q = n_0 P$, then the solutions to $Q = nP$ are the integers $n = n_0 + is$ with $i \in \mathbb{Z}$. (See Exercise 5.9.)

This means that the value of $\log_P(Q)$ is really an element of $\mathbb{Z}/s\mathbb{Z}$, i.e., $\log_P(Q)$ is an integer modulo s, where s is the order of P. For concreteness we could set $\log_P(Q)$ equal to n_0. However the advantage of defining the values to be in $\mathbb{Z}/s\mathbb{Z}$ is that the elliptic discrete logarithm then satisfies

$$\log_P(Q_1 + Q_2) = \log_P(Q_1) + \log_P(Q_2) \qquad \text{for all } Q_1, Q_2 \in E(\mathbb{F}_p). \tag{5.4}$$

Notice the analogy with the ordinary logarithm $\log(\alpha\beta) = \log(\alpha) + \log(\beta)$ and the discrete logarithm for $\mathbb{F}_p^*$ (cf. Remark 2.2). The fact that the discrete logarithm for $E(\mathbb{F}_p)$ satisfies (5.4) means that it respects the addition law when the group $E(\mathbb{F}_p)$ is mapped to the group $\mathbb{Z}/s\mathbb{Z}$. We say that the map $\log_P$ defines a *group homomorphism* (cf. Exercise 2.13)

$$\log_P : E(\mathbb{F}_p) \longrightarrow \mathbb{Z}/s\mathbb{Z}.$$

Example 5.15. Consider the elliptic curve

$$E : Y^2 = X^3 + 8X + 7 \quad \text{over } \mathbb{F}_{73}.$$

The points $P = (32, 53)$ and $Q = (39, 17)$ are both in $E(\mathbb{F}_{73})$, and it is easy to verify (by hand if you're patient and with a computer if not) that

$$Q = 11P, \qquad \text{so} \qquad \log_P(Q) = 11.$$

Similarly, $R = (35, 47) \in E(\mathbb{F}_{73})$ and $S = (58, 4) \in E(\mathbb{F}_{73})$, and after some computation we find that they satisfy $R = 37P$ and $S = 28P$, so

$$\log_P(R) = 37 \qquad \text{and} \qquad \log_P(S) = 28.$$

Finally, we mention that $\#E(\mathbb{F}_{73}) = 82$, but P satisfies $41P = \mathcal{O}$. Thus P has order $41 = 82/2$, so only half of the points in $E(\mathbb{F}_{73})$ are multiples of P. For example, $(20, 65)$ is in $E(\mathbb{F}_{73})$, but it does not equal a multiple of P.

5.3.1 The Double-and-Add Algorithm

It appears to be quite difficult to recover the value of n from the two points P and $Q = nP$ in $E(\mathbb{F}_p)$, i.e., it is difficult to solve the ECDLP. We will say more about the difficulty of the ECDLP in later sections. However, in order to use the function

$$\mathbb{Z} \longrightarrow E(\mathbb{F}_p), \qquad n \longmapsto nP,$$

for cryptography, we need to efficiently compute nP from the known values n and P. If n is large, we certainly do not want to compute nP by computing $P, 2P, 3P, 4P, \ldots$.

The most efficient way to compute nP is very similar to the method that we described in Section 1.3.2 for computing powers $a^n \pmod N$, which we needed for Diffie–Hellman key exchange (Section 2.3) and for the ElGamal and RSA public key cryptosystems (Sections 2.4 and 3.2). However, since the operation on an elliptic curve is written as addition instead of as multiplication, we call it "double-and-add" instead of "square-and-multiply."

The underlying idea is the same as before. We first write n in binary form as

$$n = n_0 + n_1 \cdot 2 + n_2 \cdot 4 + n_3 \cdot 8 + \cdots + n_r \cdot 2^r \qquad \text{with } n_0, n_1, \ldots, n_r \in \{0, 1\}.$$

Input. Point $P \in E(\mathbb{F}_p)$ and integer $n \geq 1$.
1. Set $Q = P$ and $R = \mathcal{O}$.
2. Loop while $n > 0$.
  3. If $n \equiv 1 \pmod 2$, set $R = R + Q$.
  4. Set $Q = 2Q$ and $n = \lfloor n/2 \rfloor$.
  5. If $n > 0$, continue with loop at Step **2**.
6. Return the point R, which equals nP.

Table 5.3: The double-and-add algorithm for elliptic curves

(We also assume that $n_r = 1$.) Next we compute the following quantities:

$$Q_0 = P, \quad Q_1 = 2Q_0, \quad Q_2 = 2Q_1, \quad \dots \quad , Q_r = 2Q_{r-1}.$$

Notice that Q_i is simply twice the previous Q_{i-1}, so

$$Q_i = 2^i P.$$

These points are referred to as 2-power multiples of P, and computing them requires r doublings. Finally, we compute nP using at most r additional additions,

$$nP = n_0 Q_0 + n_1 Q_1 + n_2 Q_2 + \cdots + n_r Q_r.$$

We'll refer to the addition of two points in $E(\mathbb{F}_p)$ as a *point operation*. Thus the total time to compute nP is at most $2r$ point operations in $E(\mathbb{F}_p)$. Notice that $n \geq 2^r$, so it takes no more than $2\log_2(n)$ point operations to compute nP. This makes it feasible to compute nP even for very large values of n. We have summarized the double-and-add algorithm in Table 5.3.

Example 5.16. We use the Double-and-Add Algorithm as described in Table 5.3 to compute nP in $E(\mathbb{F}_p)$ for

$$n = 947, \qquad E : Y^2 = X^3 + 14X + 19, \qquad p = 3623, \qquad P = (6, 730).$$

The binary expansion of n is

$$n = 947 = 1 + 2 + 2^4 + 2^5 + 2^7 + 2^8 + 2^9.$$

The step by step calculation, which requires nine doublings and six additions, is given in Table 5.4. The final result is $947P = (3492, 60)$. (The n column in Table 5.4 refers to the n used in the algorithm described in Table 5.3.)

Remark 5.17. There is an additional technique that can be used to further reduce the time required to compute nP. The idea is to write n using sums and differences of powers of 2. The reason that this is advantageous is because there are generally fewer terms, so fewer point additions are needed to compute nP. It is important to observe that subtracting two points on an elliptic curve is as

Step i	n	$Q = 2^i P$	R
0	947	$(6, 730)$	$\mathcal{O}$
1	473	$(2521, 3601)$	$(6, 730)$
2	236	$(2277, 502)$	$(2149, 196)$
3	118	$(3375, 535)$	$(2149, 196)$
4	59	$(1610, 1851)$	$(2149, 196)$
5	29	$(1753, 2436)$	$(2838, 2175)$
6	14	$(2005, 1764)$	$(600, 2449)$
7	7	$(2425, 1791)$	$(600, 2449)$
8	3	$(3529, 2158)$	$(3247, 2849)$
9	1	$(2742, 3254)$	$(932, 1204)$
10	0	$(1814, 3480)$	$(3492, 60)$

Table 5.4: Computing $947 \cdot (6, 730)$ on $Y^2 = X^3 + 14X + 19$ modulo 3623

easy as adding them, since $-(x, y) = (x, -y)$. This is rather different from $\mathbb{F}_p^*$, where computing a^{-1} takes significantly more time than it takes to multiply two elements.

An example will help to illustrate the idea. We saw in Example 5.16 that $947 = 1+2+2^4+2^5+2^7+2^8+2^9$, so it takes 15 point operations (9 doublings and 6 additions) to compute $947P$. But if we instead write

$$947 = 1 + 2 - 2^4 - 2^6 + 2^{10},$$

then we can compute

$$947P = P + 2P - 2^4P - 2^6P + 2^{10}P$$

using 10 doublings and 4 additions, for a total of 14 point operations. Writing a number n as a sum of positive and negative powers of 2 is called a *ternary expansion of* n.

How much savings can we expect? Suppose that n is a large number and let $k = \lfloor \log n \rfloor + 1$. In the worst case, if n has the form $2^k - 1$, then computing nP using a binary expansion of n requires $2k$ point operations (k doublings and k additions), since

$$2^k - 1 = 1 + 2 + 2^2 + \cdots + 2^{k-1}.$$

But if we allow ternary expansions, then we prove below (Proposition 5.18) that computing nP never requires more than $\frac{3}{2}k + 1$ point operations ($k + 1$ doublings and $\frac{1}{2}k$ additions).

This is the worst case scenario, but it's also important to know what happens on average. The binary expansion of a random number has approximately the same number of 1's and 0's, so for most n, computing nP using the binary expansion of n takes about $\frac{3}{2}k$ steps (k doublings and $\frac{1}{2}k$ additions). But if we

allow sums and differences of powers of 2, then one can show that most n have an expansion with $\frac{2}{3}$ of the terms being 0. So for most n, we can compute nP in about $\frac{4}{3}k+1$ steps ($k+1$ doublings and $\frac{1}{3}k$ additions).

Proposition 5.18. *Let n be a positive integer and let $k = \lfloor \log n \rfloor + 1$, which means that $2^k > n$. Then we can always write*

$$n = u_0 + u_1 \cdot 2 + u_2 \cdot 4 + u_3 \cdot 8 + \cdots + u_k \cdot 2^k \tag{5.5}$$

with $u_0, u_1, \ldots, u_k \in \{-1, 0, 1\}$ and at most $\frac{1}{2}k$ of the u_i nonzero.

Proof. The proof is essentially an algorithm for writing n in the desired form. We start by writing n in binary,

$$n = n_0 + n_1 \cdot 2 + n_2 \cdot 4 + \cdots + n_{k-1} \cdot 2^{k-1} \qquad \text{with } n_0, \ldots, n_{k-1} \in \{0, 1\}.$$

Working from left to right, we look for the first occurrence of two or more consecutive nonzero n_i coefficients. For example, suppose that

$$n_s = n_{s+1} = \cdots = n_{s+t-1} = 1 \qquad \text{and} \qquad n_{s+t} = 0$$

for some $t \geq 1$. In other words, the quantity

$$2^s + 2^{s+1} + \cdots + 2^{s+t-1} + 0 \cdot 2^{s+t} \tag{5.6}$$

appears in the binary expansion of n. We observe that

$$2^s + 2^{s+1} + \cdots + 2^{s+t-1} + 0 \cdot 2^{s+t} = 2^s(1 + 2 + 4 + \cdots + 2^{t-1}) = 2^s(2^t - 1),$$

so we can replace (5.6) with

$$-2^s + 2^{s+t}.$$

Repeating this procedure, we end up with an expansion of n of the form (5.5) in which no two consecutive u_i are nonzero. (Note that although the original binary expansion went up to only 2^{k-1}, the new expansion might go up to 2^k.) □

5.3.2 How hard is the ECDLP?

The collision algorithms described in Section 4.4 are easily adapted to any group, for example to the group of points $E(\mathbb{F}_p)$ on an elliptic curve. In order to solve $Q = nP$, Eve chooses random integers $j_1, \ldots, j_r$ and $k_1, \ldots, k_r$ between 1 and p and makes two lists of points:

$$\begin{aligned} &\text{List \#1.} \qquad j_1P,\ j_2P,\ j_3P, \ldots,\ j_rP, \\ &\text{List \#2.} \qquad k_1P + Q,\ k_2P + Q,\ k_3P + Q, \ldots,\ k_rP + Q. \end{aligned}$$

As soon as she finds a match (collision) between the two lists, she is done, since if she finds $j_uP = k_vP + Q$, then $Q = (j_u - k_v)P$ provides the solution.

As we saw in Section 4.4, if r is somewhat larger than $\sqrt{p}$, say $r \approx 3\sqrt{p}$, then there is a very good chance that there will be a collision.

This naive collision algorithm requires quite a lot of storage for the two lists. However, it is not hard to adapt Pollard's ρ method from Section 4.5 to devise a storage-free collision algorithm with a similar running time. (See Exercise 5.12.) In any case, there are certainly algorithms that solve the ECDLP for $E(\mathbb{F}_p)$ in $\mathcal{O}(\sqrt{p}\,)$ steps.

We have seen that there are much faster ways to solve the discrete logarithm problem for $\mathbb{F}_p^*$. In particular, the index calculus described in Section 3.8 has a subexponential running time, i.e., the running time is $\mathcal{O}(p^\epsilon)$ for every $\epsilon > 0$. The principal reason that elliptic curves are used in cryptography is the fact that there are no index calculus algorithms known for the ECDLP, and indeed, there are no general algorithms known that solve the ECDLP in fewer than $O(\sqrt{p}\,)$ steps. In other words, despite the highly structured nature of the group $E(\mathbb{F}_p)$, the fastest known algorithms to solve the ECDLP are no better than the generic algorithm that works equally well to solve the discrete logarithm problem in any group. This fact is sufficiently important that it bears highlighting.

The fastest known algorithm to solve ECDLP in $E(\mathbb{F}_p)$ takes approximately $\sqrt{p}$ steps.

Thus the ECDLP appears to be much more difficult than the DLP. Recall, however, there are some primes p for which the DLP in $\mathbb{F}_p^*$ is comparatively easy. For example, if $p-1$ is a product of small primes, then the Pohlig–Hellman algorithm (Theorem 2.32) gives a quick solution to the DLP in $\mathbb{F}_p^*$. In a similar fashion, there are some elliptic curves and some primes for which the ECDLP in $E(\mathbb{F}_p)$ is comparatively easy. We discuss some of these special cases, which must be avoided in the construction of secure cryptosystems, in Section 5.9.1.

5.4 Elliptic curve cryptography

It is finally time to apply elliptic curves to cryptography. We start with the easiest application, Diffie–Hellman key exchange, which involves little more than replacing the discrete logarithm problem for the finite field $\mathbb{F}_p$ with the discrete logarithm problem for an elliptic curve $E(\mathbb{F}_p)$. We then describe an elliptic analogue of the ElGamal public key cryptosystem.

5.4.1 Elliptic Diffie–Hellman key exchange

Alice and Bob agree to use a particular elliptic curve $E(\mathbb{F}_p)$ and a particular point $P \in E(\mathbb{F}_p)$. Alice chooses a secret integer n_A and Bob chooses a secret

Public Parameter Creation	
A trusted party chooses and publishes a (large) prime p, an elliptic curve E over $\mathbb{F}_p$, and a point P in $E(\mathbb{F}_p)$.	
Private Computations	
Alice	**Bob**
Chooses a secret integer n_A. Computes the point $Q_A = n_A P$.	Chooses a secret integer n_B. Computes the point $Q_B = n_B P$.
Public Exchange of Values	
Alice sends Q_A to Bob $\longrightarrow$	Q_A
Q_B $\longleftarrow$	Bob sends Q_B to Alice
Further Private Computations	
Alice	**Bob**
Computes the point $n_A Q_B$.	Computes the point $n_B Q_A$.
The shared secret value is $n_A Q_B = n_A(n_B P) = n_B(n_A P) = n_B Q_A$.	

Table 5.5: Diffie–Hellman key exchange using elliptic curves

integer n_B. They compute the associated multiples

$$\overbrace{Q_A = n_A P}^{\text{Alice computes this}} \quad \text{and} \quad \overbrace{Q_B = n_B P}^{\text{Bob computes this}},$$

and they exchange the values of Q_A and Q_B. Alice then uses her secret multiplier to compute $n_A Q_B$, and Bob similarly computes $n_B Q_A$. They now have the shared secret value

$$n_A Q_B = (n_A n_B)P = n_B Q_A,$$

which they can use as a key to communicate privately via a symmetric cipher. Table 5.5 summarizes elliptic Diffie–Hellman key exchange.

Example 5.19. Alice and Bob decide to use elliptic Diffie–Hellman with the following prime, curve, and point:

$$p = 3851, \qquad E : Y^2 = X^3 + 324X + 1287, \qquad P = (920, 303) \in E(\mathbb{F}_{3851}).$$

Alice and Bob choose respective secret values $n_A = 1194$ and $n_B = 1759$, and then

$$\begin{aligned} &\text{Alice computes} \quad Q_A = 1194P = (2067, 2178) \in E(\mathbb{F}_{3851}), \\ &\text{Bob computes} \quad Q_B = 1759P = (3684, 3125) \in E(\mathbb{F}_{3851}). \end{aligned}$$

Alice sends Q_A to Bob and Bob sends Q_B to Alice. Finally,

$$\begin{aligned} &\text{Alice computes} \quad n_A Q_B = 1194(3684, 3125) = (3347, 1242) \in E(\mathbb{F}_{3851}), \\ &\text{Bob computes} \quad n_B Q_A = 1759(2067, 2178) = (3347, 1242) \in E(\mathbb{F}_{3851}). \end{aligned}$$

Bob and Alice have exchanged the secret point $(3347, 1242)$. As will be explained in Remark 5.20, they should discard the y-coordinate and treat only the value $x = 3347$ as a secret shared value.

One way for Eve to discover Alice and Bob's secret is to solve the ECDLP

$$nP = Q_A,$$

since if Eve can solve this problem, then she knows n_A and can use it to compute $n_A Q_B$. Of course, there might be some other way for Eve to compute their secret without actually solving the ECDLP. The precise problem that Eve needs to solve is the elliptic analogue of the Diffie–Hellman problem described on page 67.

Definition. Let $E(\mathbb{F}_p)$ be an elliptic curve over a finite field and let $P \in E(\mathbb{F}_p)$. The *Elliptic Curve Diffie–Hellman Problem* is the problem of computing the value of $n_1 n_2 P$ from the known values of $n_1 P$ and $n_2 P$.

Remark 5.20. Elliptic Diffie–Hellman key exchange requires Alice and Bob to exchange points on an elliptic curve. A point Q in $E(\mathbb{F}_p)$ consists of two coordinates $Q = (x_Q, y_Q)$, where x_Q and y_Q are elements of the finite field $\mathbb{F}_p$, so it appears that Alice must send Bob two numbers in $\mathbb{F}_p$. However, those two numbers modulo p do not contain as much information as two arbitrary numbers, since they are related by the formula

$$y_Q^2 = x_Q^3 + Ax_Q + B \qquad \text{in } \mathbb{F}_p.$$

Note that Eve knows A and B, so if she can guess the correct value of x_Q, then there are only two possible values for y_Q, and in practice it is not too hard for her to actually compute the two values of y_Q.

There is thus little reason for Alice to send both coordinates of Q_A to Bob, since the y-coordinate contains so little additional information. Instead, she sends Bob only the x-coordinate of Q_A. Bob then computes and uses one of the two possible y-coordinates. If he happens to choose the "correct" y, then he is using Q_A, and if he chooses the "incorrect" y (which is the negative of the correct y), then he is using $-Q_A$. In any case, Bob ends up computing one of

$$\pm n_B Q_A = \pm(n_A n_B)P.$$

Similarly, Alice ends up computing one of $\pm(n_A n_B)P$. Then Alice and Bob use the x-coordinate as their shared secret value, since that x-coordinate is the same regardless of which y they use.

Example 5.21. Alice and Bob decide to exchange another secret value using the same public parameters as in Example 5.19:

$$p = 3851, \qquad E : Y^2 = X^3 + 324X + 1287, \qquad P = (920, 303) \in E(\mathbb{F}_{3851}).$$

However, this time they want to send fewer bits to one another. Alice and Bob respectively choose new secret values $n_A = 2489$ and $n_B = 2286$, and as before,

Alice computes $Q_A = n_A P = 2489(920, 303) = (593, 719) \in E(\mathbb{F}_{3851})$,
Bob computes $Q_B = n_B P = 2286(920, 303) = (3681, 612) \in E(\mathbb{F}_{3851})$.

However, rather than sending both coordinates, Alice sends only $x_A = 593$ to Bob and Bob sends only $x_B = 3681$ to Alice.

Alice substitutes $x_B = 3681$ into the equation for E and finds that

$$y_B^2 = x_B^3 + 324x_B + 1287 = 3681^3 + 324 \cdot 3681 + 1287 = 997.$$

(Recall that all calculations are performed in $\mathbb{F}_{3851}$.) Alice needs to compute a square root of 997 modulo 3851. This is not hard to do, especially for primes satisfying $p \equiv 3 \pmod 4$, since Proposition 2.27 tells her that $b^{(p+1)/4}$ is a square root of b modulo p. So Alice sets

$$y_B = 997^{(3851+1)/4} = 997^{963} \equiv 612 \pmod{3851}.$$

It happens that she gets the same point $Q_B = (x_B, y_B) = (3681, 612)$ that Bob used, and she computes $n_A Q_B = 2489(3681, 612) = (509, 1108)$.

Similarly, Bob substitutes $x_A = 593$ into the equation for E and takes a square root,

$$\begin{aligned} y_A^2 &= x_A^3 + 324x_A + 1287 = 593^3 + 324 \cdot 593 + 1287 = 927, \\ y_A &= 927^{(3851+1)/4} = 927^{963} \equiv 3132 \pmod{3851}. \end{aligned}$$

Bob then uses the point $Q'_A = (593, 3132)$, which is not Alice's point Q_A, to compute $n_B Q'_A = 2286(593, 3132) = (509, 2743)$. Bob and Alice end up with points that are negatives of one another in $E(\mathbb{F}_p)$, but that is all right, since their shared secret value is the x-coordinate $x = 593$, which is the same for both points.

5.4.2 Elliptic ElGamal public key cryptosystem

It is easy to create a direct analogue of the ElGamal public key cryptosystem described in Section 2.4. Briefly, Alice and Bob agree to use a particular prime p, elliptic curve E, and point $P \in E(\mathbb{F}_p)$. Alice chooses a secret multiplier n_A and publishes the point $Q_A = n_A P$ as her public key. Bob's plaintext is a point $M \in E(\mathbb{F}_p)$. He chooses an integer k to be his ephemeral key and computes

$$C_1 = kP \qquad \text{and} \qquad C_2 = M + kQ_A.$$

He sends the two points (C_1, C_2) to Alice, who computes

$$C_2 - n_A C_1 = (M + kQ_A) - n_A(kP) = M + k(n_A P) - n_A(kP) = M$$

to recover the plaintext. The elliptic ElGamal public key cryptosystem is summarized in Table 5.6.

In principle, the elliptic ElGamal cryptosystem works fine, but there are some practical difficulties.

Public Parameter Creation	
A trusted party chooses and publishes a (large) prime p, an elliptic curve E over $\mathbb{F}_p$, and a point P in $E(\mathbb{F}_p)$.	
Alice	**Bob**
Key Creation	
Chooses a private key n_A. Computes $Q_A = n_A P$ in $E(\mathbb{F}_p)$. Publishes the public key Q_A.	
Encryption	
	Chooses plaintext $M \in E(\mathbb{F}_p)$. Chooses an ephemeral key k. Uses Alice's public key Q_A to compute $C_1 = kP \in E(\mathbb{F}_p)$. and $C_2 = M + kQ_A \in E(\mathbb{F}_p)$. Sends ciphtertext (C_1, C_2) to Alice.
Decryption	
Computes $C_2 - n_A C_1 \in E(\mathbb{F}_p)$. This quantity is equal to M.	

Table 5.6: Elliptic ElGamal key creation, encryption, and decryption

1. There is no obvious way to attach plaintext messages to points in $E(\mathbb{F}_p)$.
2. The elliptic ElGamal cryptosystem has 4-to-1 message expansion, as compared to the 2-to-1 expansion ratio of ElGamal using $\mathbb{F}_p$. (See Remark 2.9.)

The reason that elliptic ElGamal has a 4-to-1 message expansion lies in the fact that the plaintext M is a single point in $E(\mathbb{F}_p)$. By Hasse's theorem (Theorem 5.11) there are approximately p different points in $E(\mathbb{F}_p)$, hence only about p different plaintexts. However, the ciphertext (C_1, C_2) consists of four numbers modulo p, since each point in $E(\mathbb{F}_p)$ has two coordinates.

Various methods have been proposed to solve these problems. The difficulty of associating plaintexts to points can be circumvented by choosing M randomly and using it as a mask for the actual plaintext. One such method, which also decreases message expansion, is described in Exercise 5.16.

Another natural way to improve message expansion is to send only the x-coordinates of C_1 and C_2, as was suggested for Diffie–Hellman key exchange in Remark 5.20. Unfortunately, since Alice must compute the difference $C_2 - n_A C_1$, she needs the correct values of both the x-and y-coordinates of C_1 and C_2. (Note that the points $C_2 - n_A C_1$ and $C_2 + n_A C_1$ are quite different!) However, the x-coordinate of a point determines the y-coordinate up to change of sign, so Bob can send one extra bit, for example

$$\text{Extra bit} = \begin{cases} 0 & \text{if } 0 \le y < \frac{1}{2}p, \\ 1 & \text{if } \frac{1}{2}p < y < p \end{cases}$$

(See Exercise 5.15.) In this way, Bob needs to send only the x-coordinates of C_1 and C_2, plus two extra bits. This idea is sometimes referred to as *point compression*.

5.5 The evolution of public key cryptography

The invention of RSA in the late 1970s catapulted the problem of factoring large integers into prominence, leading to improved factorization methods such as the quadratic and number field sieves described in Section 3.7. In 1984, Hendrik Lenstra Jr. circulated a manuscript describing a new factorization method using elliptic curves. Lenstra's algorithm [71], which we describe in Section 5.6, is an elliptic analogue of Pollard's $p-1$ factorization algorithm (Section 3.5) and exploits the fact that the number of points in $E(\mathbb{F}_p)$ varies as one chooses different elliptic curves. Although less efficient than sieve methods for the factorization problems that occur in cryptography, Lenstra's algorithm helped introduce elliptic curves to the cryptographic community.

The importance of factorization algorithms for cryptography is that they are used to break RSA and other similar cryptosystems. In 1985, Neal Koblitz and Victor Miller independently proposed using elliptic curves to create cryptosystems. They suggested that the elliptic curve discrete logarithm problem might be more difficult than the classical discrete logarithm problem modulo p. Thus Diffie–Hellman key exchange and the ElGamal public key cryptosystem, implemented using elliptic curves as described in Section 5.4, might require smaller keys and run more efficiently than RSA because one could use smaller numbers.

Koblitz [62] and Miller [79] each published their ideas as academic papers, but neither of them pursued the commercial aspects of elliptic curve cryptography. Indeed, at the time, there was virtually no research on the ECDLP, so it was difficult to say with any confidence that the ECDLP was indeed significantly more difficult than the classical DLP. However, the potential of what became known as elliptic curve cryptography (ECC) was noted by Scott Vanstone and Ron Mullin, who had started a cryptographic company called Certicom in 1985. They joined with other researchers in both academia and the business world to promote ECC as an alternative to RSA and ElGamal.

All was not smooth sailing. For example, during the late 1980s, various cryptographers proposed using so-called supersingular elliptic curves for added efficiency, but in 1990, the MOV algorithm (see Section 5.9.1) showed that supersingular curves are vulnerable to attack. Some saw this as an indictment of ECC as a whole, while others pointed out that RSA also has weak instances that must be avoided, e.g., RSA must avoid using numbers that can be easily factored by Pollard's $p-1$ method.