Contents

5 Elliptic Curves in Cryptography

In this chapter, we will introduce elliptic curves and describe how they are used in cryptography. Elliptic curves have a long and interesting history and arise in a wide range of contexts in mathematics. The study of elliptic curves involves elements from most of the major disciplines of mathematics: algebra, geometry, analysis, number theory, topology, and even logic. Elliptic curves appear in the proofs of many deep results in mathematics: for example, they are a central ingredient in the proof of Fermat's Last Theorem, which states that there are no positive integer solutions to the equation $x^n + y^n = z^n$ for any integer $n \geq 3$.

Our goals are fairly modest in comparison, so we will begin by outlining the basic algebraic and geometric properties of elliptic curves and motivate the "addition law". We will then study the behavior of elliptic curves modulo p: ultimately, there is a fairly strong analogy between the structure of the points on an elliptic curve modulo p and the integers modulo n. Our goal is to explore this analogy and then to use it to "convert" cryptosystems and factorization algorithms that rely on modular arithmetic to ones that rely on elliptic curves when possible: ultimately, we will construct elliptic curve analogues of ElGamal encryption, Diffie-Hellman key exchange, and ElGamal signatures.

5.1 Elliptic Curves and the Addition Law

- In this section, we will outline the basic features of elliptic curves that we will use in subsequent sections to do cryptography.

5.1.1 Cubic Curves, Weierstrass Form, Singular and Nonsingular Curves

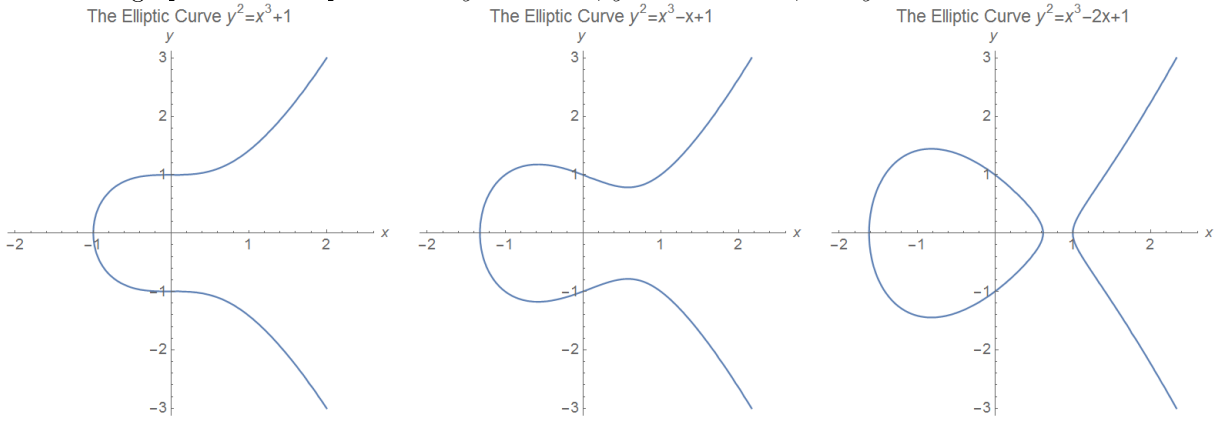
- In elementary coordinate geometry, one often begins by studying the behavior of lines in the plane, which have the general equation $ax + by + c = 0$. In trigonometry (or precalculus), one often studies more general quadratic curves (the conic sections) having the general equation $ax^2 + bxy + cy^2 + dx + ey + f = 0$.
    - In each case, we often perform simple algebraic manipulations and changes of variable to put the equations into a more standard form.
    - For example, if $b \neq 0$, we can rewrite the equation $ax + by + c = 0$ as $y = (-a/b)x + (-c/b)$, which for $m = -a/b$ and $b_1 = -c/b$ has the more familiar form $y = mx + b_1$.

- Similarly, if $a \neq 0$, we can perform a change of variable $x_1 = y + (b/(2a))x$ in the equation $ax^2 + bxy + cy^2 + dx + ey + f = 0$ to remove the cross term bxy: we eventually obtain an equation of the form $ax_1^2 + c_1y^2 + d_1x_1 + e_1y + f_1 = 0$ for new coefficients c_1, d_1, e_1, f_1.
- We can then complete the square in both x_1 and y (again, assuming certain coefficients are nonzero) by setting $x_2 = x_1 + d_1/(2a_1)$ and $y_2 = y + e_1/(2c_1)$. Eventually we will obtain an equation having the much simpler $ax_2^2 + c_1y_2^2 + f_2 = 0$.
- If we abuse notation by dropping the subscripts, we see that essentially every conic can be put into the form $ax^2 + cy^2 + f = 0$ after changing coordinates.

- Our goal is to study cubic curves in the plane, which have the general form $ax^3 + bx^2y + cxy^2 + dy^3 + ex^2 + fxy + gy^2 + hx + iy + j = 0$.
  - Like in the case of quadratic curves above, we can perform a series of changes of variable to reduce the general form to a simpler one.
  - We will not give the full details of the procedure, as it is rather complicated.
  - Instead, we will summarize matters by saying that as long as the equation is actually cubic (i.e., it is not the case that all of a, b, c, d are zero), then the general equation above can always be transformed using rational changes of variable into one of the form $y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$, for appropriate coefficients a_1, a_2, a_3, a_4, a_6.

- Definition: An elliptic curve E over a field K is a curve having an equation of the form $y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$, for appropriate coefficients a_1, a_2, a_3, a_4, a_6 in K. This expression is called the Weierstrass form of E.
  - Note: We will generally restrict our attention to the situation where K is either the field of real numbers, the field of complex numbers, or the field of integers modulo p. (We will not assume any knowledge about any fields other than these three.)
  - This expression is not the simplest possible one: we can simplify it by completing the square in y and completing the cube in x.
  - Explicitly, if we set $y' = y + (a_1/2)x + (a_3/2)$ and $x' = x + (a_2/3)$, we can reduce the Weierstrass equation above to one of the form $(y')^2 = (x')^3 + A(x') + B$.
  - An elliptic curve having an equation of the form $y^2 = x^3 + Ax + B$ is sometimes said to be in "reduced" Weierstrass form.
  - This reduced form is much more amenable for computations, and (in fact) it is nearly unique: the only change of variables that preserves it is one of the form $x = u^2x'$, $y = u^3y'$ for some nonzero u, from which we see that $A = u^4A'$ and $B = u^6B'$.

- Here are the graphs of the elliptic curves $y^2 = x^3 + 1$, $y^2 = x^3 - x + 1$, and $y^2 = x^3 - 2x + 1$:

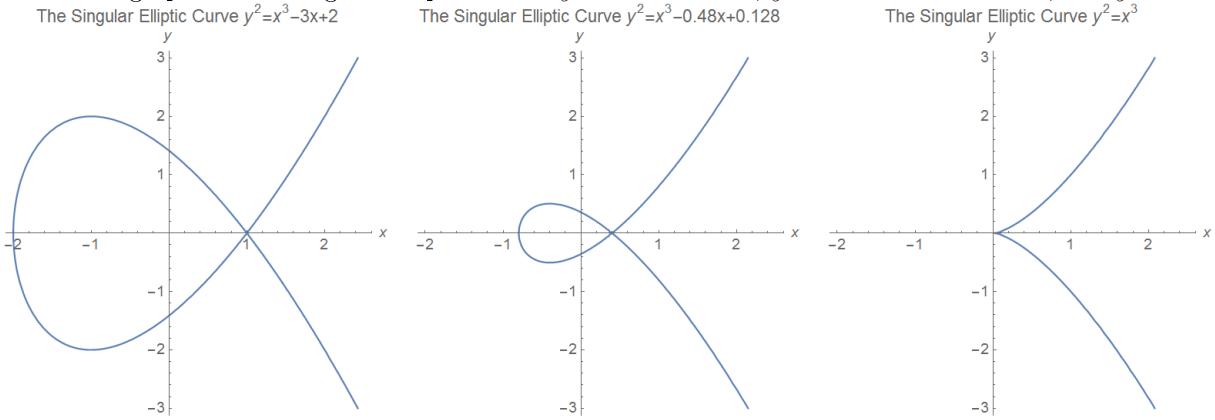


  - Note that elliptic curves are not ellipses! For completeness, the reason for the similar name is that if one wants to compute the arclength of an ellipse (an elliptic integral), a few changes of variable will transform the resulting integral into one of the general form $\displaystyle\int \frac{1}{\sqrt{x^3 + Ax + B}}\,dx$. Upon setting $y = \sqrt{x^3 + Ax + B}$, we see that this elliptic integral is rather naturally related to the curve $y^2 = x^3 + Ax + B$.

- In general, we can see that the graph of an elliptic curve $y^2 = x^3 + Ax + B$ will always be symmetric about the x-axis, since if (x, y) satisfies the equation then so does $(x, -y)$.
- By using this observation and invoking a result known as the implicit function theorem, it can be shown that the graph of an elliptic curve will have either one or two components depending on the values of the coefficients: it will have two components when the polynomial $x^3 + Ax + B$ has three distinct real roots, and it will have one component otherwise.
- Notice also that the tangent line at each crossing of the x-axis is vertical for each curve above. Using implicit differentiation, we can compute $y' = \dfrac{3x^2 + A}{2y}$: thus, we see that $y' = \infty$ when y is zero, provided that as $3x^2 + A$ is not also zero. This behavior can only occur when $x^3 + Ax + B$ has a root in common with its derivative $3x^2 + A$, which is in turn equivalent to saying that $x^3 + Ax + B$ has a double root.

- Definition: If the polynomial $x^3 + Ax + B$ has a repeated root, we say that the elliptic curve $y^2 = x^3 + Ax + B$ is singular. Otherwise (if the roots are distinct) we say the elliptic curve is nonsingular. A curve is singular if and only if its discriminant $\Delta = -16(4A^3 + 27B^2)$ is zero.
  - The second statement follows from the observations above: the polynomial $x^3 + Ax + B$ has a repeated root if and only if it has a root in common with its derivative $3x^2 + A$. This occurs precisely when $x^2 = -A/3$, from which we see that $x(2A/3) + B = 0$ so $x = -3B/(2A)$: then substituting for x yields $\Delta = 0$ almost immediately.
  - Remark: The presence of the constant -16 is superfluous here, but there is also a definition of Δ in terms of the original coefficients a_1, a_2, a_3, a_4, a_6 for a general Weierstrass form. To avoid having denominators in that expression, we end up needing an extra factor of -16 in the one we gave above.

- Here are the graphs of the singular elliptic curves $y^2 = x^3 - 3x + 2$, $y^2 = x^3 - 0.48x + 0.128$, and $y^2 = x^3$:



  - The singular point (i.e., the point where the curve is nondifferentiable) on the first two curves is where the curve crosses itself. This type of singularity is known as a node, and will occur when the polynomial $x^3 + Ax + B$ has a double root.
  - The singular point on the third curve is the cusp at the origin $(0, 0)$. This type of singularity will occur when the polynomial $x^3 + Ax + B$ has a triple root (which can only happen when $A = B = 0$).

- In general, singular elliptic curves tend to have unusual properties relative to nonsingular curves. We will therefore exclude singular elliptic curves and speak only of nonsingular elliptic curves from this point on.

5.1.2 The Addition Law

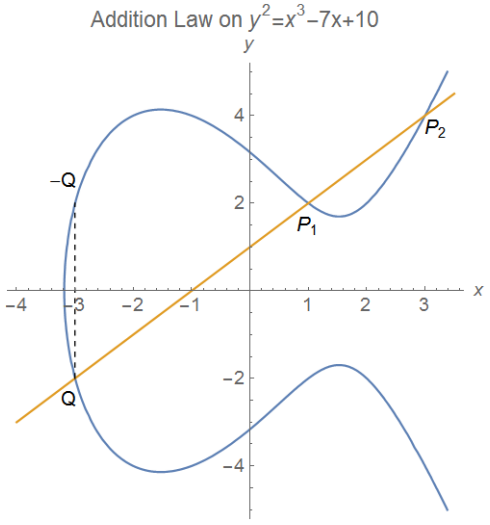
- The key property of elliptic curves that make them so useful is that, if we have two points that lie on the curve, we can use them to construct a third point on the curve.
  - Explicitly, suppose $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$ are two distinct points on an elliptic curve E: $y^2 = x^3 + Ax + B$.
  - Draw the line through P_1 and P_2: we claim that this line L must intersect E in a third point Q.

- To see this, suppose the line through P_1 and P_2 has equation $y = mx + b$. (We are tacitly excluding the possibility that the line is vertical, but we will come back to this case in a moment.)
- Then the intersection points between L and E are the solutions to the system $y = mx+b$ and $y^2 = x^3 + Ax+B$. Equivalently, we must solve $(mx+b)^2 = x^3+Ax+B$, or $x^3+(-m^2)x^2+(A-2mb)x+(B-b^2) = 0$.
- However, we already know that this cubic has the two roots $x = x_1$ and $x = x_2$, so it must have a third root: this gives us the third point Q we wanted.

- Once we construct a third point on an elliptic curve this way, we might try to find more points.
  - If we try this procedure directly using our points P_1, P_2, and Q, however, we will not get anywhere: the line through any of these two points intersects the elliptic curve at the other point.
  - However, we can also exploit the vertical symmetry of the curve to make new points: if $P = (x, y)$ lies on the curve, then the point $-P = (x, -y)$ also lies on the curve.
  - If we combine these two procedures, we can often generate many points on the curve starting from just two.

- Definition (Group Law I): If P_1 and P_2 are two distinct points on the elliptic curve $E : y^2 = x^3 + Ax + B$, let $Q = (x', y')$ be the third intersection point of E with the line L joining P_1 and P_2. We define the sum $P_1 + P_2$ to be the point $-Q = (x', -y')$.
  - Important Note: The sum $P_1 + P_2$ is *not* the pointwise coordinate sum of P_1 and P_2!
  - It is not immediately clear why we define the sum of two points to be the reflection of Q rather than Q itself. This will become clearer in a moment.
  - Note that if we attempt to add two points which are vertical reflections of one another on the graph of $y^2 = x^3 + Ax + B$, the resulting line will not intersect the curve again.
  - To remedy this, we declare that the curve also includes a point at ∞ (which we denote simply as ∞) that we consider as lying on any vertical line.
  - There are other important technical reasons for this declaration, most of which we do not have the background to discuss at this level.

- Example: Given the points $P_1 = (1, 2)$ and $P_2 = (3, 4)$ on the elliptic curve $y^2 = x^3 - 7x + 10$, find the sums $P_1 + P_2$ and $(P_1 + P_2) + P_2$.
  - It is easy to verify that both points lie on the curve. Here is a plot of the curve and the line $y = x + 1$ through the two points:



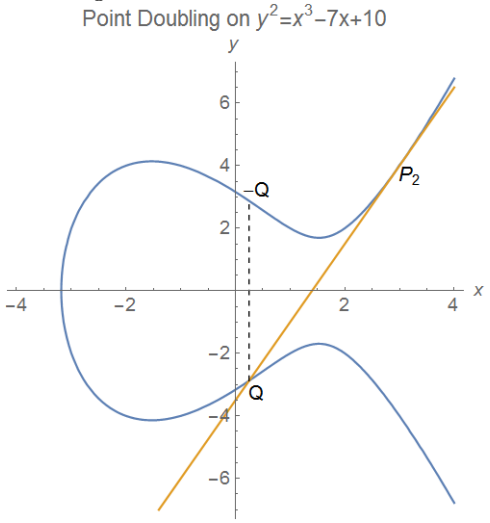
  - Now we find the exact coordinates of Q.
  - The point lies on the intersection of $y = x + 1$ and $y^2 = x^3 - 7x + 10$, so $(x + 1)^2 = x^3 - 7x + 10$.

- This equation is equivalent to $x^3 - x^2 - 9x + 9 = 0$, which factors as $(x-1)(x-3)(x+3) = 0$. Then the x-coordinate of Q is -3 so $Q = (-3, -2)$.
- Thus, the sum $P_1 + P_2$ is the vertical reflection of Q, which is $\boxed{(-3, 2)}$.
- To find the sum $(P_1 + P_2) + P_2$ we perform a similar procedure: the line through $P_1 + P_2$ and P_2 has equation $y = \frac{1}{3}x + 3$.
- Then we must solve $(\frac{1}{3}x + 3)^2 = x^3 - 7x + 10$, or $x^3 - \frac{1}{9}x^2 - 9x + 1 = 0$.
- Factoring yields $(x - \frac{1}{9})(x+3)(x-3) = 0$, so $Q' = (\frac{1}{9}, \frac{82}{27})$, and thus $(P_1 + P_2) + P_2 = \boxed{(\frac{1}{9}, -\frac{82}{27})}$.

- Now that we have defined addition, a natural question is whether we can add a point to itself.
  - It is straightforward to see from our definition that if P_1 and P_2 are distinct points, then $P_1 + P_2$ is a continuous function of the coordinates of the points.
  - We could therefore define the addition $P + P$ to be the limit as $P_1 \to P$ of sums $P + P_1$. Geometrically, the lines used in the construction also have a limit as $P \to P_1$: they approach the tangent line to the curve E at the point P.
  - Thus, the most natural way to define $P + P$ is to let L be the tangent line to E at P, and then take Q to be the third point of intersection of L with E.

- Definition (Group Law II): If P is any point on the elliptic curve $E : y^2 = x^3 + Ax + B$, let $Q = (x', y')$ be the third intersection point of E with the tangent line L to E at P. We define the sum $P + P$ to be the point $-Q = (x', -y')$.

- Example: Given the points $P_1 = (1, 2)$ and $P_2 = (3, 4)$ on the elliptic curve $y^2 = x^3 - 7x + 10$, find the sums $P_2 + P_2$ and $(P_1 + P_2) + P_2$.
  - By differentiating implicitly, we see that the tangent line to E at P_2 has slope $\frac{5}{2}$, and so the equation is $y = \frac{5}{2}x - \frac{7}{2}$.
  - Here is a plot of the curve and the tangent line at P_2:



  - The point Q lies on the intersection of $y = \frac{5}{2}x - \frac{7}{2}$ and $y^2 = x^3 - 7x + 10$, so $(\frac{5}{2}x - \frac{7}{2})^2 = x^3 - 7x + 10$.
  - This equation is equivalent to $x^3 - \frac{25}{4}x^2 + \frac{21}{2}x - \frac{9}{4} = 0$, which factors as $(x - \frac{1}{4})(x-3)(x-3) = 0$.
  - Then the x-coordinate of Q is $1/4$ so $Q = (\frac{1}{4}, -\frac{23}{8})$, and so $P_2 + P_2 = \boxed{(\frac{1}{4}, \frac{23}{8})}$.

- To find the sum $P_1 + (P_2 + P_2)$ we then find the sum of $P_1 = (1, 2)$ with $(\frac{1}{4}, \frac{23}{8})$. The line through these points is $y = -\frac{7}{6}x + \frac{19}{6}$.
- Then we must solve $(-\frac{7}{6}x + \frac{19}{6})^2 = x^3 - 7x + 10$, which has solutions $x = \frac{1}{9}, \frac{1}{4}, 1$.
- Then $Q' = (\frac{1}{9}, \frac{82}{27})$, and thus $P_1 + (P_2 + P_2) = \boxed{(\frac{1}{9}, -\frac{82}{27})}$.

- Note that in the previous two examples, we computed $(P_1 + P_2) + P_2 = (\frac{1}{9}, -\frac{82}{27}) = P_1 + (P_2 + P_2)$, and so we see in this case that the addition law is actually associative. Much more is true:
- Theorem (Group Law): If K is any field and E is any elliptic curve defined over K, then for any points P, P_1, P_2, and P_3 on E, the following are true:
  1. The addition law is commutative: $P_1 + P_2 = P_2 + P_1$.
  2. The addition law is associative: $(P_1 + P_2) + P_3 = P_1 + (P_2 + P_3)$.
  3. The point at ∞ is a two-sided identity: $P + \infty = P = \infty + P$.
  4. The point P has a two-sided inverse $-P$: $P + (-P) = \infty = (-P) + P$.
  - A more concise way of phrasing this statement is to say that the set of points on E (including the point at ∞) forms an abelian group.
  - We will give arguments for an elliptic curve of the form $y^2 = x^3 + Ax + B$, but the theorem holds in full generality for any elliptic curve.
  - Proof (1): The first part is immediate from the geometric definition we have given since the line used in computing $P_1 + P_2$ and $P_2 + P_1$ is the same in each case.
  - Proof (2): This part, which is the only nontrivial result in this theorem, can be done with a lengthy numerical computation using explicit formulas for the addition law (see below). We will omit the details.
  - Proof (3): Consider the sum $P + \infty$. The line passing through P and ∞ is the vertical line through P which also intersects E at the point $-P$. Then by the geometric definition, $P + \infty = -(-P) = P$.
  - Proof (4): Consider the sum $P + (-P)$. The line passing through P and $-P$ is a vertical line, so the other point on it is ∞. The reflection of ∞ is also ∞, so $P + (-P) = \infty$.
- For convenience in doing numerical computations, we will also write down the general formula for the addition law on any curve:
- Proposition (Explicit Group Law): Let $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$ be points on the elliptic curve $E : y^2 = x^3 + Ax + B$. Then $P_1 + P_2 = (x_3, y_3)$ where $x_3 = m^2 - x_1 - x_2$ and $y_3 = -m(x_3 - x_1) - y_1$, with $m = \begin{cases} (y_2 - y_1)/(x_2 - x_1) & \text{if } P_1 \neq P_2 \\ (3x_1^2 + A)/(2y_1) & \text{if } P_1 = P_2 \end{cases}$. If m is infinite, then $P_1 + P_2 = \infty$.
  - We will remark that the addition formula is rational, in the sense that the result is always a rational function of the inputs. In particular, the sum of two points whose coordinates lie in a field K will also lie in K.
  - We will also remark that there are formulas for the addition law on a more general elliptic curve $y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$, but we will not need them.
  - Proof: If $P_1 \neq P_2$ then the line joining P_1 and P_2 has equation $y - y_1 = m(x - x_1)$ where $m = \frac{y_2 - y_1}{x_2 - x_1}$.
  - We therefore obtain the equation $(mx - mx_1 + y_1)^2 = x^3 + Ax + B$, which has the form $x^3 - m^2x^2 + Cx + D = 0$ for appropriate constants C and D.
  - The polynomial $x^3 - m^2x^2 + Cx + D$ must factor as $(x - x_1)(x - x_2)(x - x_3)$, so upon multiplying out we see that $x_1 + x_2 + x_3 = m^2$. This yields the stated value of x_3, and then $y_3 = m(x_3 - x_1) + y_1$ (where we have multiplied by -1 to account for the vertical reflection).
  - If $P_1 = P_2$ then everything is the same, except instead m is the slope of the tangent line at P_1. By implicit differentiation, we see that $2yy' = 3x^2 + A$ so $m = \frac{3x_1^2 + A}{2y_1}$ here, as claimed.

5.1.3 Elliptic Curves Modulo p, Orders of Points

- We have primarily dealt with elliptic curves over the real numbers, but an important part of the general theory requires studying elliptic curves modulo p, where p is prime. We will take $p \geq 5$ to be a prime throughout the remainder of this section.
  - We will discuss elliptic curves modulo nonprime integers when we discuss factorization algorithms.
- All of our analysis of elliptic curves carries into this setting essentially verbatim: in particular, the properties of the addition law and the algebraic formulas remain the same, though we must rely on algebra rather than geometric intuition.
  - The only possible difficulty is that if we want to work in a field of "characteristic 2" (in which $2 = 0$) or "characteristic 3" (in which $3 = 0$), we will need to use the general Weierstrass form $y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$ rather than the reduced Weierstrass form $y^2 = x^3 + Ax + B$.
  - As we showed earlier, an elliptic curve $y^2 = x^3 + Ax + B$ is nonsingular modulo p precisely when its discriminant $\Delta = -16(4A^3 + 27B^2)$ is nonzero modulo p.
  - In particular, we can see that a curve of this form will always be singular modulo 2.
  - More generally, if we have any elliptic curve curve, the primes p for which the curve is singular mod p (the primes of "bad reduction") are precisely the primes dividing the discriminant Δ.
- Example: If $P_1 = (1, 3)$ and $P_2 = (0, 2)$ on the elliptic curve $y^2 = x^3 + 4x + 4$ modulo 5, find $P_1 + P_2$ and $P_1 + P_1$.
  - We simply apply the appropriate formulas: adding $Q_1 = (x_1, y_1)$ to $Q_2 = (x_2, y_2)$ produces (x_3, y_3) where $x_3 = m^2 - x_1 - x_2$ and $y_3 = -m(x_3 - x_1) - y_1$, and $m = \begin{cases} (y_2 - y_1)/(x_2 - x_1) & \text{if } Q_1 \neq Q_2 \\ (3x_1^2 + A)/(2y_1) & \text{if } Q_1 = Q_2 \end{cases}$.
  - With $(x_1, y_1) = (1, 3)$ and $(x_2, y_2) = (0, 2)$ we obtain $m = \dfrac{2-3}{0-1} = 1$, so $x_3 = 0$ and $y_3 = -1(0-1)-3 = 3$, so $P_1 + P_2 = \boxed{(0, 3)}$.
  - Likewise, with $(x_1, y_1) = (x_2, y_2) = (1, 3)$ we obtain $m = \dfrac{3+4}{2 \cdot 3} = 2$, so $x_3 = 2$ and $y_3 = -2(2-1)-3 = 0$, so $P_1 + P_1 = \boxed{(2, 0)}$.
- Since there are only finitely many pairs of numbers modulo p, any elliptic curve E will have only finitely many points modulo p, and so we can in principle write them all down (at least if p is small).
  - Usually, the easiest procedure for doing this is to try plugging in each possible value of x and then try to compute the square root of $x^3 + Ax + B$ to find the value of y.
  - In our count, we also include the point at ∞ on our list.
- Example: Construct an addition table for the (nonsingular) elliptic curve $y^2 = x^3 + 4x + 4$ modulo 3.
  - First, we find all the points by plugging in each of the possible x and computing the necessary square roots. We obtain

x	0	1	2
$x^3 + 4x + 4$	1	0	2
y	± 1	0	n/a

    and so there are 4 points on the curve modulo 3: $(0, 1)$, $(0, 2)$, $(1, 0)$, and ∞.
  - We can now compute all of the sums using the algebraic formulas:

$+$	∞	$(0,1)$	$(0,2)$	$(1,0)$
∞	∞	$(0,1)$	$(0,2)$	$(1,0)$
$(0,1)$	$(0,1)$	$(1,0)$	∞	$(0,2)$
$(0,2)$	$(0,2)$	∞	$(1,0)$	$(0,1)$
$(1,0)$	$(1,0)$	$(0,2)$	$(0,1)$	∞

- Example: Verify that the elliptic curve $y^2 = x^3 + 4x + 4$ is nonsingular mod p and then find all the points on the curve mod p, where $p = 5$, 7, 11, and 13.
  - For the nonsingularity part, we compute the discriminant $\Delta = -16 \cdot 688 = -2^8 \cdot 43$. Since none of 5, 7, 11, 13 divide the discriminant, the curve is nonsingular for each of these moduli.
  - To count the points, we plug in each possible value of x mod p and then try to compute the square root of $x^3 + Ax + B$.
  - Modulo 5, we obtain

x	0	1	2	3	4
$x^3 + 4x + 4$	4	4	0	3	4
y	± 2	± 2	0	n/a	± 2

    and so there are $\boxed{8}$ points modulo 5: $(0, 2)$, $(0, 3)$, $(1, 2)$, $(1, 3)$, $(2, 0)$, $(4, 2)$, $(4, 3)$, and ∞.
  - Modulo 7, we obtain

x	0	1	2	3	4	5	6
$x^3 + 4x + 4$	4	2	6	1	0	2	6
y	± 2	± 3	n/a	± 1	0	± 3	n/a

    and so there are $\boxed{10}$ points modulo 7: $(0, 2)$, $(0, 5)$, $(1, 3)$, $(1, 4)$, $(3, 1)$, $(3, 6)$, $(4, 0)$, $(5, 3)$, $(5, 4)$, and ∞.
  - Modulo 11, we obtain

x	0	1	2	3	4	5	6	7	8	9	10
$x^3 + 4x + 4$	4	9	9	10	7	6	2	1	9	10	10
y	± 2	± 3	± 3	n/a	n/a	n/a	n/a	± 1	± 3	n/a	n/a

    and so there are $\boxed{11}$ points modulo 11: $(0, \pm 2)$, $(1, \pm 3)$, $(2, \pm 3)$, $(7, \pm 1)$, $(8, \pm 3)$, and ∞.
  - Modulo 13, we obtain

x	0	1	2	3	4	5	6	7	8	9	10	12	13
$x^3 + 4x + 4$	4	9	7	4	6	6	10	11	2	2	4	1	12
y	± 2	± 3	n/a	± 2	n/a	n/a	± 6	n/a	n/a	n/a	± 2	± 1	± 5

    and so there are $\boxed{15}$ points modulo 13: $(0, \pm 2)$, $(1, \pm 3)$, $(3, \pm 2)$, $(6, \pm 6)$, $(10, \pm 2)$, $(12, \pm 1)$, $(13, \pm 5)$, and ∞.

- Notice that the number of points on the elliptic curve E modulo p in the example above was fairly close to p for each value we tested. It turns out that this is no accident:

- Theorem (Hasse): Let E be a nonsingular elliptic curve defined over a finite field with q elements. Then the number of points $N_q(E)$ on E whose entries are in K satisfies $|N_q(E) - q - 1| \leq 2\sqrt{q}$.
  - Remark: A stronger result holds for singular curves: the number of points on a singular elliptic curve (including the singular point itself) is always either p, $p+1$, or $p+2$ depending on the type of singularity.
  - Remark: When p is a prime, each of the possible values of $N_p(E)$ satisfying the inequality $|N_p(E) - p - 1| \leq 2\sqrt{p}$ actually does occur as the number of points on some elliptic curve mod p.
  - We will not prove this result, which is usually known as the Hasse bound, as it requires significantly more advanced methods than those we will develop. However, we can give a bit of motivation by finding the number of points we should expect to be on an elliptic curve modulo p.
  - For each of the p possible values of x, there are either 2, 1, or 0 possible values of y, according to whether x is a nonzero square, zero, or a nonsquare. It can be shown that (when p is an odd prime) there are $(p-1)/2$ nonzero squares modulo p, so the expected number of values of y for any particular x is $\frac{1}{p}\left[2 \cdot \frac{p-1}{2} + 1 \cdot 1 + 0 \cdot \frac{p-1}{2}\right] = \frac{1}{p}[p - 1 + 1] = 1$.
  - Since there are p possible x, the expected number of points (x, y) is $p \cdot 1 = p$. Together with the point at ∞, this gives $p + 1$ points on the curve E.
  - Trivially, we can see that $1 \leq N_p(E) \leq 2p + 1$, each value of x contributes at most 2 values of y, and the point at ∞ always counts. We can rearrange this inequality to read $|N_p(E) - p - 1| \leq p$.
  - Hasse's theorem is then a strengthening of this inequality: it says that the actual number of points on the curve is comparatively close to the expected number of points, with the upper bound p on the difference replaced with the (comparatively much smaller) bound $2\sqrt{p}$.

- Our goal now is to set up a rough analogy between the structure of the points on an elliptic curve modulo p under addition and the units modulo n under multiplication.
    - Ultimately, the similarities between the structure of points on an elliptic curve modulo p and the integers modulo n stem from the fact that the set of points on an elliptic curve modulo p under addition is a finite abelian group, as is the set of units modulo n.
- Our first goal is to define the order of a point on an elliptic curve. To do this we will use the addition operation on the curve:
- Definition: Suppose E is an elliptic curve defined over a field K, and P is a point on E. For any positive integer k, we define the point kP to be the sum $\underbrace{P+P+\cdots+P}_{k \text{ terms}}$, and we also define $(-k)P$ to be the additive inverse $-(kP)$ along with $0P = \infty$. The smallest positive k for which $kP = \infty$ is then called the order of P; if no such k exists, then we say P has infinite order. A point of finite order is called a torsion point.
    - Compare this definition to the one in modular arithmetic: the order of a unit u modulo m is the smallest $k > 0$ such that $u^k \equiv 1 \pmod{m}$.
    - Note that kP is well-defined because the addition law is associative: it does not matter the order in which we perform the additions. Likewise, we can see more or less immediately that $(a+b)P = aP + bP$ for any integers a and b.
    - Over the real or complex numbers, most points on an elliptic curve will have infinite order. (More precisely, the set of torsion points is countably infinite, while the set of all points is uncountable.)
    - As we will show, however, on an elliptic curve modulo p all points have finite order.
- Example: Find the order of the point $P = (1, 3)$ on the elliptic curve $E : y^2 = x^3 + 4x + 4$ modulo 5.
    - We simply compute the multiples of P using the addition law repeatedly.
    - We obtain $2P = P + P = (2, 0)$, $3P = 2P + P = (1, 2)$, $4P = 3P + P = \infty$.
    - Since $4P$ is the smallest multiple of P that gives the point ∞, the order of P is $\boxed{4}$.
- We can compute large multiples of a particular point using successive doubling, in analogy to the procedure of successive squaring:
- Algorithm (Successive Doubling): To compute kP, first find the binary expansion of $k = b_j b_{j-1} \cdots b_0$. Then compute the multiples $2P$, $4P$, $8P$, ... , $2^j P$ by using the doubling part of the addition law. Finally, compute $kP = \sum_{\substack{0 \le i \le j \\ b_i = 1}} 2^{b_i} P$ using the addition law.
    - We can speed this procedure up a bit by also using subtractions: unlike with modular arithmetic, where it is comparatively expensive to compute inverses, if $P = (x, y)$ then we have the trivial formula $-P = (x, -y)$.
    - We will also observe that this procedure works for any elliptic curve, not just an elliptic curve modulo p. The only issue is that large multiples of a typical point will usually grow very complicated over an infinite field.
- Orders of points on an elliptic curve share many of the same properties as orders of units modulo an integer m, and the proofs of these results are also essentially the same.
- Proposition: If P is a point of order k on the elliptic curve E and $mP = \infty$, then k divides m.
    - Proof: Suppose $mP = \infty$ and write $m = qk + r$ where $0 \le r < k$.
    - We then have $rP = mP + (-qk)P = mP + (-q)(kP) = \infty + (-q)\infty = \infty + \infty = \infty$.
    - Since $rP = \infty$ and $0 \le r < k$, the only possibility is to have $r = 0$: otherwise this would contradict the minimality of k. Thus $m = qk$ so k divides m.

- Proposition: If P is a point on the elliptic curve E such that $mP = \infty$, but $(m/q)P \neq \infty$ for any prime divisor q of m, then P has order m.
  - Proof: Suppose the order of P is k. Then since $mP = \infty$, by the previous proposition we conclude that k divides m.
  - If $k < m$, then there must be some prime q in the prime factorization of m that appears to a strictly lower power in the factorization of k: then k divides m/q.
  - But then $(m/q)P = \infty$ since m/q is a multiple of k, but this is contrary to the given information. Thus $m = k$ so P has order m.
- Theorem (Point Orders on Elliptic Curves Mod p): If E is an elliptic curve modulo a prime p and N is the number of points on E modulo p, then $NP = \infty$. In particular, the order of P divides N.
  - This result is an analogue of Euler's theorem. It is a corollary of a more general result of group theory known as Lagrange's theorem, which states that the order of any element of a group divides the number of elements in the group.
  - In our case, we can adapt the proof of Euler's theorem with minimal difficulty.
  - Proof: Suppose the points on E are $Q_1, Q_2, \cdots, Q_N$ and consider the points $Q_1+P, Q_2+P, \cdots, Q_N+P$: we claim that they are simply the points $Q_1, Q_2, \cdots, Q_N$ again (possibly in a different order).
  - Since there are N points listed and they all lie on the curve E, it is enough to verify that they are all distinct.
  - So suppose $Q_i + P = Q_j + P$. Then we can write $Q_i = Q_i + \infty = Q_i + (P + (-P)) = (Q_i + P) + (-P) = (Q_j + P) + (-P) = Q_j + (P + (-P)) = Q_j + \infty = Q_j$, where we used associativity and the properties of ∞ and inverses. (Morally, we simply subtracted P from both sides.)
  - Thus the points $Q_1 + P, Q_2 + P, \cdots, Q_N + P$ are simply $Q_1, Q_2, \cdots, Q_N$ in some order. Adding up all the terms then yields $(Q_1 + P) + \cdots + (Q_N + P) = Q_1 + \cdots + Q_N$, and upon rearranging and subtracting $Q_1 + \cdots + Q_N$ from both sides (in the same way as above), we obtain $NP = \infty$ as desired.
  - The second statement follows immediately from $NP = \infty$ and the propositions above.
- Example: Show that the point $P = (1, 3)$ has order 15 on the elliptic curve $E : y^2 = x^3 + 4x + 4$ modulo 13.
  - It is a straightforward check that $15P = \infty$ using successive doubling: we compute $2P = (12, 8)$, $4P = (6, 6)$, $8P = (0, 11)$, $16P = (1, 3)$. Then $15P = 16P - P = (1, 3) - (1, 3) = \infty$.
  - Furthermore, we can compute $3P = 2P + P = (3, 2)$ and $5P = 4P + P = (10, 2)$.
  - Since neither of these quantities is ∞, we conclude that the order of P must be $\boxed{15}$.
- If we can compute the orders of some points on E, we can often use that information in conjunction with the Hasse bound to determine the number of points on E without actually computing them all.
  - In the above example, we exhibited a point of order 15 on the elliptic curve $E : y^2 = x^3 + 4x + 4$ modulo 13. Thus, by our results on orders, the number of points on E must be a multiple of 15.
  - By the Hasse bound, the number of points on E must satisfy $|N - 14| \leq 2\sqrt{13}$, yielding the inequality $6.78 \leq N \leq 21.22$. The only multiple of 15 in this range is 15 itself, so E must have exactly 15 points.

5.2 Factorization with Elliptic Curves

- Now that we have a reasonably good analogy between modular multiplication and the points on an elliptic curve modulo p under addition, we can use the analogy to create a factorization algorithm using elliptic curves that is based off of Pollard's $(p-1)$-algorithm. These ideas were first proposed by H. Lenstra in 1985.
  - In Pollard's $(\rho - 1)$-algorithm, the basic idea is that if $n = pq$ and we choose a random integer a, then the order of a modulo p is likely to differ from the order of a modulo q.
  - Thus, if the order of a mod p is k and is larger than k mod q, $a^k \equiv 1 \pmod{p}$ but $a^k \not\equiv 1 \pmod{q}$, so that $\gcd(a^k - 1, n) = p$.

- Let us now try to construct an appropriate analogy with elliptic curves:
  - Again, suppose $n = pq$ is a product of two primes, and suppose we choose a (nonsingular) elliptic curve $E : y^2 = x^3 + Ax + B$ over the integers along with a point P on the curve.
  - The order of P on E_p, the reduction of E modulo p, is unlikely to be exactly equal to the order of P on E_q, the reduction of E modulo q.
  - If the order of P on E_p is k and the order of P on E_q is larger than k, then $kP = \infty$ on E_p but $kP \neq \infty$ on E_q.
  - Now the question arises: how can we detect this behavior? In Pollard's $(p-1)$-algorithm, we performed all our calculations modulo n, so it seems we should do the same thing here.
  - Thus, we do all of our computations on the curve E_n, the reduction of the curve E modulo n, using the addition law formulas defined over the rational numbers reduced modulo n.
  - Assuming that this reduction is well-defined, the addition law will still obey all of the requirements we put on it (namely, it will be commutative, associative, have an identity ∞, and have inverses).
  - However, the addition law formulas require a division when computing the slope of the line, and if this slope requires dividing by a nonzero number that is not invertible mod n, then we will not be able to evaluate the result. (If we were dividing by zero itself, then we would simply obtain a slope of ∞: however, there is no sensible way to interpret a slope of $\frac{1}{2}$ modulo 6.)
  - This is precisely what we want: it is saying that the slope of the line is ∞ modulo one of the prime divisors of n, but not ∞ modulo the other. Then to find the nontrivial divisor of n, we simply take the gcd of the problematic denominator with n.
  - Another way to interpret this idea is using the Chinese remainder theorem: a point (x, y) lies on E_n if and only if it lies on the curve $E_p : y^2 = x^3 + Ax + B$ modulo p and the curve $E_q : y^2 = x^3 + Ax + B$ modulo q.
  - Thus, the points on E_n can equivalently be thought of as pairs of points (P, Q) of points on E_p and E_q. We are then seeking to detect when a multiple of a pair (P, Q) is ∞ in one coordinate but not in the other.

- Example: Examine what happens when trying to add the point $P_1 = (1, 3)$ to the point $P_2 = (15, 4)$ on the elliptic curve $E_{21} : y^2 = x^3 + 4x + 4$ modulo 21, and when doubling the point P_1.
  - To find $P_1 + P_2$ we compute the slope of the line: it is $\frac{4-3}{15-1} = \frac{1}{14}$. However, this rational number is not defined modulo 21, since 14 is not relatively prime to 21. In this case, we see that $\gcd(21, 14) = 7$ is a proper divisor of 21.
  - Similarly, if we try to compute $P_1 + P_1$, the slope of the tangent line is $\frac{3(1)^2+4}{2 \cdot 3} = \frac{7}{6}$, which is again not defined modulo 21 since 6 is not relatively prime to 21. In this case, we see that $\gcd(21, 6) = 3$ is a proper divisor of 21.
  - Ultimately, what is happening in the first case is that $P_1 + P_2 = \infty \pmod 7$ but $P_1 + P_2 \neq \infty \pmod 3$. In the second case, $2P_1 = \infty \pmod 3$ but $2P_1 \neq \infty \pmod 7$.

- To implement this procedure to factor integers in a reasonable way requires a bit more care, but again we can take guidance from Pollard's $(p-1)$-algorithm.
  - Searching through all possible k in Pollard's $(p-1)$-algorithm is very inefficient. To speed things up, we observe that it is unnecessary to find the exact order of a mod p: any multiple of it will suffice, as long as that multiple is not also divisible by the order of a mod q.
  - A reasonably efficient procedure is to evaluate $\gcd(a^{d!} - 1, n)$ for $1 \leq d \leq M$ (for some choice of bound M) until we obtain a gcd that is larger than 1.
  - In the elliptic curve analogy, we should therefore try computing $(d!)P$ on an elliptic curve $E_n : y^2 = x^3 + Ax + B$ modulo n for $1 \leq d \leq M$, and seeing if we obtain a denominator that has a nontrivial gcd with n in the denominator. If we do, then we get a factorization of n.

- The only remaining question is how to choose an ellipt0.85ic curve E along with a point P. An easy way to generate a pair (E, P) is to choose the coordinates of $P = (x_0, y_0)$ along with the value A first, and then set $B = y_0^2 - x_0^3 - Ax_0$.

- Lenstra's algorithm is simply a reformulation of these ideas:

- Algorithm (Lenstra's Factorization Algorithm): Suppose n is composite. Choose a bound M, a point $P = (x_0, y_0)$, and an integer A. Let E be the elliptic curve $E : y^2 = x^3 + Ax + B$ modulo n. Set $Q_1 = P$ and for $2 \leq j \leq M$, define $Q_j = jQ_{j-1}$ (computed on E_n). If at any stage of the computation the point Q_j cannot be computed, due to a necessary division by a denominator d which is not 0 modulo n but which is not invertible modulo n, then $\gcd(d, n)$ is a proper divisor of n. If a divisor is not found and Q_M is not ∞, increase the value of M and continue the computation. Otherwise, if $Q_M = \infty$, repeat the procedure with a new choice of P and A.

  - We will remark that the curve E can be singular, as long as P is not the singular point on the curve. (By "singular" we mean singular mod p or mod q, which is equivalent to saying that the discriminant Δ has a common prime divisor with n.)
  - However, choosing E to be a singular curve is not optimal, because (as it turns out) the algorithm will essentially reduce either to Pollard's $(p-1)$-algorithm or trial division according to the type of singularity.

- Example: Use Lenstra's factorization algorithm to find a divisor of the integer $n = 170999$ using the point $P = (1, 4)$ on the elliptic curve $E : y^2 = x^3 + 4x + 11$.

  - We simply compute the points Q_j successively using the recursion $Q_1 = P$, $Q_j = jQ_{j-1}$ on the elliptic curve E modulo n until we obtain a problematic denominator.

j	1	2	3	4	5
Q_j	$(1, 4)$	$(109545, 75144)$	$(81282, 86818)$	$(100818, 143145)$	$(152033, 116998)$
Factor?	no	no	no	no	no
j	6	7	8	9	10
Q_j	$(87978, 17295)$	$(104368, 99929)$	$(126411, 167685)$	$(79623, 108587)$	–
Factor?	no	no	no	no	557

  - In this case, attempting to compute $10Q_9$ will require dividing by a denominator that is not relatively prime to n.
  - The exact details of the computation will depend on the method used to compute $10Q_9$, but successive doubling will yield $2Q_9 = (147257, 97701)$ and $8Q_9 = (160625, 116187)$, and attempting to add these two points will require using a line with slope $m = \dfrac{116187 - 97701}{160625 - 147257} = \dfrac{18486}{13368}$, and $\gcd(13368, 170999) = 557$.

- The elliptic curve factorization algorithm seems to work, but it is not obvious how fast it is nor how efficient it is in comparison to our other algorithms.

  - As we noted above, the factorization algorithm will succeed after M steps when the order of P on the elliptic curve E_p (i.e., E modulo p) divides $M!$, but the order of P on E_q (i.e., E modulo q) does not divide $M!$.
  - It is unlikely that these two things will occur at exactly the same value of M, so what we are really seeking is for the order of P on E_p to divide $M!$.
  - From our results on orders, we know that the order of P on E_p divides the number of points N on E_p, so we are certainly guaranteed to succeed if N divides $M!$.
  - Furthermore, by the Hasse bound, $|N - p| \leq 2\sqrt{p}$. It is in fact known that N can take any integral value in this interval, and (conjecturally) it does so according to a distribution that is not far from being uniform.
  - Thus, the elliptic curve factorization will succeed quickly as long as the prime divisors of N are all fairly small.

- Note that this is a similar criterion to that of Pollard's $(p-1)$-algorithm, which succeeds quickly as long as the prime divisors of $p-1$ are all fairly small. (An integer all of whose prime divisors are $\leq M$ is called M-smooth.)
- However, we are free to make different choices for the elliptic curve E, each of which will give a different random integer that is near p. As long as one of the curves we choose is M-smooth, we will obtain the factorization of n.
- Thus, elliptic curve factorization is much more versatile than Pollard's $(p-1)$-algorithm, because in the latter if $p-1$ has a large prime divisor then we are simply out of luck, whereas with elliptic curve factorization if N has a large prime divisor then we can simply switch to a different curve. (Of course, we will generally not know the exact value of N, so we would instead switch curves if we have spent a long time computing and not gotten any results yet.)

- Another advantage to using several curves is that the computations can be completely parallelized (i.e., they can be run on separate processors), since the point operations on different curves have nothing to do with one another.
  - It is a rather nontrivial analytic number theory problem to determine the appropriate heuristic for the density of integers in the "Hasse interval" $|N-p| \leq 2\sqrt{p}$ that are M-smooth, which is needed in order to estimate how many curves should be used in order to search for the factorization and to estimate the value of M that should be used.
  - We will not give the details of this computation, but the approximately optimal pairs (M,k) for the bound M and the number of curves k are roughly $(2000, 25)$ for 15-digit prime divisors, $(10000, 100)$ for 20-digit prime divisors, and $(50000, 300)$ for 25-digit prime divisors.
  - Overall, if one computes the total time requirement with optimal choices for the parameters, Lenstra's elliptic curve algorithm can factor an integer n in a total of approximately $e^{\sqrt{2}(\ln p)^{1/2}(\ln\ln p)^{1/2}}$ steps, where p is the smallest prime divisor of n.
  - This number of steps is bounded above by $e^{(\ln n)^{1/2}(\ln\ln n)^{1/2}}$, and so the elliptic curve factorization has roughly the same asymptotic speed as the quadratic sieve.
  - In practice, due to the fact that elliptic curve operations are slower than modular exponentiations, Lenstra's algorithm becomes slower than the sieve methods for integers exceeding 60 digits or so, and is slower than Pollard's ρ-algorithm for numbers under 30 digits.
  - However, the elliptic curve method is much more efficient at finding comparatively small divisors (around 30 digits or less) of large integers than the sieve methods are.
  - To factor a large integer that is not expected to be the product of only large primes (e.g., an RSA modulus), one often uses some combination of trial division, the Pollard $(p-1)$ algorithm, and the Pollard-ρ algorithm to search for small factors (under 15 digits or so) and Lenstra's algorithm to search for factors of medium size (15-30 digits). Then one uses a sieve method to factor the remaining integer, which will be a product only of large primes.

- Finally, we will mention that there are several improvements and optimizations that can be made to Lenstra's original algorithm.
  - The largest computational overhead in Lenstra's algorithm is computing the point multiplications. There are various ways to arrange the arithmetic operations in such a way that fewer computations are needed: in particular, it is possible to use both additions and subtractions when doing successive doubling (since computing the inverse of a point is essentially free). Furthermore, by using different models for elliptic curves other than the reduced Weierstrass form $y^2 = x^3 + Ax + B$, further savings are possible.
  - It is also possible to choose the elliptic curve $y^2 = x^3 + Ax + B$ in such a way that it is still essentially random modulo n, but is guaranteed to have points of various small orders (such as 12). Such restrictions would then necessarily imply that the number of points on the curve is divisible by 12, marginally reducing the size of potential large prime divisors of N.
  - There are also "second stage" methods for Lenstra's algorithm (initially proposed by R. Brent[1]) that apply a procedure similar to Pollard's ρ-algorithm to try to find a factorization.

[1] See the paper "Some Integer Factorization Algorithms using Elliptic Curves" by R.P. Brent, Australian Comp. Sci. Comm. 8 (1986), available on arXiv:1004.3366.

- Briefly, if we compute $Q = (M!)P$ where M is fairly large but do not obtain a factorization, then (barring anything particularly weird) the order of Q on E_p will not have any prime divisors less than M.
- Suppose that Q ends up having prime order on E_p. If we can find two "random" multiples of $k_1Q = (x_1, y_1)$ and $k_2Q = (x_2, y_2)$ that are equal on E_p but not on E_q, then $\gcd(y_2 - y_1, n)$ will be equal to p.
- It would not be efficient to search directly for such multiples: instead, we could generate k such points using a "random function" (e.g., one that doubles a point half the time, and doubles then adds Q the other half of the time). Then we would compute the product $d = \prod_{1 \leq i < j \leq k}(y_i - y_j)$, evaluate $\gcd(d, n)$, and hope that it gives p.
- As with the analysis of Pollard's ρ-algorithm, we would want to take $k \approx 2\sqrt{r}$ where r is the expected order of Q. Efficiently evaluating the product modulo n is a rather nontrivial task (since it contains about $k^2/2$ terms), but there are methods for doing this, and overall it is possible to glean some small time savings over Lenstra's original algorithm.

5.3 Elliptic Curve Cryptography

- We will now develop several cryptographic protocols relying on the addition law on an elliptic curve. These will include a public-key cryptosystem based on ElGamal encryption, a key-exchange protocol based on Diffie-Hellman key exchange, and a digital signature algorithm.

5.3.1 Encoding Plaintexts on Elliptic Curves, Quadratic Residues

- Our first task is to explain how to encode messages as points on elliptic curves.
- Unlike with cryptosystems based on modular arithmetic, where we can simply write a message as a residue class modulo m (possibly with some kind of padding scheme to increase security), it is not quite so trivial to encode a message as a point on an elliptic curve if we specify the curve E ahead of time, as would be necessary for a public-key cryptosystem.
  - So suppose we have chosen an elliptic curve $y^2 = x^3 + Ax + B$ modulo a prime p, and wish to convert a message m into a point on the curve. We can assume that m is smaller than p, since we may break m up into pieces and send each piece separately using whatever scheme we come up with.
  - However we cannot, for example, simply convert a message m into the point (m, y) on the elliptic curve, because there may not be a value of y satisfying the equation $y^2 = m^3 + Am + B \pmod{p}$.
- To go further, we need to study the question of how to check whether a particular residue class is a square modulo p.
- Definition: If a is a residue class modulo p, we say a is a quadratic residue if there is some b such that $b^2 \equiv a \pmod{p}$. If there is no such b, then we say a is a quadratic nonresidue.
  - It is straightforward to list the quadratic residues by squaring all of the residue classes.
  - Example: Modulo 5, the quadratic residues are 0, 1, and 4, while the nonresidues are 2 and 3.
  - Example: Modulo 7, the quadratic residues are 0, 1, 4, and 2, while the nonresidues are 3, 5, and 6.
  - Example: Modulo 13, the quadratic residues are 0, 1, 4, 9, 3, 12, and 10, while the nonresidues are 2, 5, 6, 7, 8, and 11.
  - From these examples it seems that there are $(p+1)/2$ quadratic residues modulo p.
- Proposition: There are $(p + 1)/2$ quadratic residues modulo p, and they are the values 0^2, 1^2, 2^2, ... , $((p-1)/2)^2$.
  - Proof: Since p is prime, the statement $p|(a^2 - b^2)$ implies $p|(a - b)$ or $p|(a + b)$; thus, $a^2 \equiv b^2 \pmod{p}$ is equivalent to $a \equiv \pm b \pmod{p}$. In particular, we see that 0^2, 1^2, 2^2, ... , $((p-1)/2)^2$ are all distinct modulo p.
  - On the other hand, any square mod p is congruent to one of 0^2, 1^2, ... , $(p-1)^2$, and since $(p-x)^2 \equiv x^2 \pmod{p}$, the squares $((p+1)/2)^2$, ... , $(p-1)^2$ are equivalent to terms already on the list.

- There is a commonly-used notation for indicating whether a residue class is a square or nonsquare modulo p:
- Definition: If p is an odd prime, the Legendre symbol $\left(\frac{a}{p}\right)$ is defined to be 1 if a is a quadratic residue, -1 if a is a quadratic nonresidue, and 0 if $p|a$.
  - The notation for the Legendre symbol is somewhat unfortunate, since it is the same as that for a standard fraction; it is nonetheless standard. When appropriate, we may write $\left(\frac{a}{p}\right)_L$ to emphasize that we are referring to a Legendre symbol rather than a fraction.
  - Example: We have $\left(\frac{2}{7}\right) = +1$, $\left(\frac{3}{7}\right) = -1$, and $\left(\frac{0}{7}\right) = 0$, since 2 is a quadratic residue and 3 is a nonresidue modulo 7.
  - Example: We have $\left(\frac{3}{13}\right) = \left(\frac{-3}{13}\right) = +1$, and $\left(\frac{2}{15}\right) = 1$, since 3 and -3 are quadratic residues modulo 13, while 2 is not.
  - Note that the quadratic equation $x^2 \equiv a \pmod{p}$ has exactly $1 + \left(\frac{a}{p}\right)$ solutions modulo p.
- There is a simple way to detect squares mod p if we have a discrete logarithm table:
- Proposition: If p is an odd prime and u is a primitive root modulo p, then a unit a is a quadratic residue if and only if $\log_u(a)$ is even.
  - Proof: If $\log_u(a) = 2k$ is even, then for $m \equiv u^k$ we have $a = u^{2k} \equiv m^2$.
  - Conversely, suppose $a \equiv m^2 \pmod{p}$ with $m \neq 0$. Since u is a primitive root, we can write $m = u^k$ for some k: then $a = u^{2k}$, so $\log_u(a)$ is even.
- It is difficult to evaluate discrete logarithms in general, so the above proposition does not give an efficient way to determine whether a given residue class is a square. However, we can use it to find a much faster method:
- Theorem (Euler's Criterion): If p is an odd prime, then for any residue class a, it is true that $\left(\frac{a}{p}\right) = a^{(p-1)/2} \pmod{p}$.
  - Proof: If $p|a$ then both sides are zero, so now assume a is a unit modulo p and let u be a primitive root modulo p.
  - First suppose a is a quadratic residue, so that $\left(\frac{a}{p}\right) = +1$. By the proposition above, we know $a = u^{2k}$ for some integer k; then $a^{(p-1)/2} \equiv (u^{2k})^{(p-1)/2} = (u^{p-1})^k \equiv 1^k = 1 \pmod{p}$, which agrees with $\left(\frac{a}{p}\right)$.
  - Now suppose a is a quadratic nonresidue, so that $\left(\frac{a}{p}\right) = -1$. Again by the proposition above, we know $a = u^{2k+1}$ for some integer k; then we compute $a^{(p-1)/2} \equiv (u^{2k+1})^{(p-1)/2} = (u^{p-1})^k \cdot u^{(p-1)/2} \equiv u^{(p-1)/2}$.
  - Now observe that $x = u^{(p-1)/2}$ has the property that $x^2 \equiv 1 \pmod{p}$. The two solutions to this quadratic are $x \equiv \pm 1 \pmod{p}$, but $x \not\equiv 1 \pmod{p}$ since otherwise u would not be a primitive root.
  - Hence $u^{(p-1)/2} \equiv -1 \pmod{p}$, meaning that $a^{(p-1)/2} \equiv -1 \pmod{p}$ as well, and this agrees with $\left(\frac{a}{p}\right)$.
- Example: Determine whether $a = 17441$ and $b = 135690$ are quadratic residues modulo the prime $p = 239441$.
  - We simply compute $a^{(p-1)/2} \equiv a^{119720} \equiv 1 \pmod{p}$, so by Euler's criterion $\boxed{a \text{ is a quadratic residue}}$ mod p.
  - Likewise, $b^{(p-1)/2} \equiv b^{119720} \equiv -1 \pmod{p}$, so by Euler's criterion $\boxed{b \text{ is not a quadratic residue}}$ mod p.
- As one of many corollaries of Euler's criterion, we can deduce that the Legendre symbol is multiplicative:

- Corollary: For any odd prime p, the Legendre symbol modulo p is multiplicative: $\left(\frac{ab}{p}\right) = \left(\frac{a}{p}\right)\left(\frac{b}{p}\right)$. In particular, the product of two quadratic nonresidues is a quadratic residue.
  - Proof: Observe $\left(\frac{ab}{p}\right) \equiv (ab)^{(p-1)/2} \equiv a^{(p-1)/2}b^{(p-1)/2} \equiv \left(\frac{a}{p}\right)\left(\frac{b}{p}\right) \pmod{p}$.
- Finally, we will recapitulate a result from studying Rabin encryption that allows us to compute square roots modulo a prime congruent to 3 modulo 4:
- Proposition: If p is a prime congruent to 3 modulo 4 and a is a quadratic residue modulo p, then $x = a^{(p+1)/4}$ has $x^2 \equiv a \pmod{p}$.
  - Proof: Since $a = m^2 \pmod{p}$ by hypothesis and $m^{p-1} \equiv 1 \pmod{p}$ by Fermat's little theorem, we can then write $x^2 \equiv a^{(p+1)/2} \equiv m^{p+1} \equiv m^2 \equiv a \pmod{p}$.
  - We will also remark that there are fast algorithms for computing square roots of quadratic residues modulo primes congruent to 1 modulo 4, but they are more complicated.
- We can now return to the question of encoding messages on an elliptic curve $E : y^2 = x^3 + Ax + B$ modulo p, where we will now also take $p \equiv 3 \pmod{4}$.
  - From the above analysis, we would expect, based on the fact that $(p+1)/2$ of the residues modulo p are squares, that for any given x there should exist a y with $y^2 = x^3 + Ax + B$ mod p about half of the time.
  - If we try to encode a message directly as the x-coordinate of a point, we therefore should only expect to succeed about half of the time.
  - A better procedure is instead to encode a message as part of the x-coordinate of a point, and then try to choose the remaining piece of the x-coordinate in such a way that $x^3 + Ax + B$ is a quadratic residue modulo p.
  - Here is a particular scheme for doing this: if p has $r + k + 1$ bits when written in base 2, we break the message into pieces each containing r bits.
  - Then, to convert an r-bit message m, we pad the beginning m with $k + 1$ bits: a zero followed by k bits $b_1 b_2 \cdots b_k$ that can be arbitrarily chosen, and set x to be the bit string $0b_1 \cdots b_k m$.
  - We then search through the possible choices of these k bits until we find a solution y to $y^2 = x^3 + Ax + B \pmod{p}$, and pick one of the two possible values of y arbitrarily. We then perform our encryption procedure using the point (x, y) on E modulo p.
  - To recover the message m from a point (x, y), where $0 \leq x < p$ we simply compute x modulo 2^r and write the result as a bit string in base 2.
  - Ultimately, since there are 2^k possible choices for the bit string $b_1 b_2 \cdots b_k$, the probability that none of them yields a quadratic residue $x^3 + Ax + B$ is roughly $1 - 2^{-2^k}$. (Of course, the probabilities are not entirely independent, but they should be nearly so.)
  - Even if we merely take $k = 10$, this probability is already so vanishingly small that it is unlikely a problem would ever occur in practical deployment.
- Example: Encode the message $m = 13 = 1101_2$ as a point on the elliptic curve $y^2 = x^3 + 11x + 17$ modulo $p = 307$ using a message length $r = 4$ bits and a padding length of $k = 4$ bits.
  - We note that $p > 256 = 2^8$ so p has 9 bits in base 2.
  - We therefore want to search for a bit string $b_1 b_2 b_3 b_4$ such that $x = 0b_1 b_2 b_3 b_4 1101_2$ is a quadratic residue modulo 307.
  - The bit string 0000 yields the value $x = 13$, but $x^3 + 11x + 17 \equiv 208 \pmod{307}$ is a quadratic nonresidue as can be confirmed by evaluating $208^{153} \equiv -1 \pmod{307}$.
  - The bit string 0001, however, yields $x = 29$, and $x^3 + 11x + 1 \equiv 165 \pmod{307}$ is a quadratic residue as can be confirmed by evaluating $165^{153} \equiv 1 \pmod{307}$.
  - To compute the associated value of y, we then compute $x^{(p+1)/4} \equiv 29^{77} \equiv 120 \pmod{307}$. A point associated to m on the curve E is then $\boxed{(29, 120)}$.

- Of course, there are many other such points: another is the additive inverse $(29, 187)$.
- We could also have searched more randomly for possible bit strings (rather than starting at 0000 and going upward), to try to keep the procedure from being as predictable. The bit string 1110, for example, yields another possible point $(237, 209)$.
- To recover the message m, we simply extract the x-coordinate and reduce it modulo $2^4 = 16$. This yields the correct original message $13 = 1101_2$.

5.3.2 Public-Key Encryption with Elliptic Curves

- We now discuss the creation of public-key cryptosystems using elliptic curves, which was first proposed by Neal Koblitz and Victor Miller in 1985.
  - Based on our earlier discussion on how to convert messages to points on curves, we will assume throughout that we our plaintext is a point (x, y) on a given elliptic curve E.
  - We will generally work with the reduction E_p of E modulo a prime p, and N will denote the number of points on E_p.
- A natural first guess for how to create a public-key cryptosystem would be to adapt RSA or Rabin encryption to the elliptic curve setting: however, some difficulties will arise if we try to do this.
  - An RSA/Rabin-like procedure would involve roughly the following: Bob creates a public key consisting of an elliptic curve E, a prime p, and an "encryption multiplier" e.
  - If Alice wants to encrypt a plaintext message $P = (x, y)$, she computes the ciphertext $C = eP$ on E_p and sends it to Bob.
  - To decrypt a ciphertext C, Bob then computes $P = dC$ for an appropriate "decryption" multiplier d.
  - In order for everything to work properly, Bob needs $(de - 1)P = \infty$ for every possible message P. From our results on orders, this is essentially equivalent to requiring that $de \equiv 1 \pmod{N}$ where N is the number of points on E_p.
  - Thus, to compute appropriate values of d and e, Bob would need to compute the number of points on E_p.
  - As we have seen, this task is not entirely trivial, although there is a procedure known as Schoof's algorithm can compute the number of points on an elliptic curve modulo p in time approximately equal to $(\log p)^5$. (An improvement due to Elkies and Atkin can heuristically improve this result to $(\log p)^4$.)
  - Roughly speaking, the idea of Schoof's algorithm is to compute the value of N modulo enough small primes that we can find N modulo r for a value of r larger than $4\sqrt{p}$: then the Hasse bound will yield a unique possible value of N.
  - However, an immediate problem arises: in order for Eve to break the cryptosystem, it is clearly sufficient for her to compute the number of points on E_p, as she can then compute d the same way Bob does. If Eve has a computer that is at least as powerful as Bob's, then she can break the cryptosystem completely.
  - Ultimately, there does not seem to be a good way to avoid this problem.
  - Suppose we instead try to work with an elliptic curve modulo a nonprime integer $n = pq$: then addition law will not always work properly. If we ignore that particular issue, the system is essentially using a pair of points (P, Q), one on E_p and one on E_q, and an appropriate pair (e, d) can be found as a solution to the congruence $de \equiv 1 \pmod{N_p N_q}$.
  - However, in this case, Eve would be able to break the system by factoring n, since she could then compute the values N_p and N_q using Schoof's algorithm above. The usage of elliptic curves here does not add to the security, and merely serves to complicate everything.
- Instead of trying to use the difficulty of inverting modular exponentiation (which is only hard when the modulus is composite), we should instead try to build systems that rely on the difficulty of computing discrete logarithms, which is a more natural problem for elliptic curves modulo a prime p. We will therefore describe a procedure for an elliptic curve version of ElGamal encryption.

- First, Bob must create his public key.
  - To do this, he chooses an elliptic curve E, a prime p, and a point Q_a on E whose order is large.
  - Ideally, Bob should choose the point Q_a to have an order whose value is a large prime roughly equal to the number of points on the curve E_p, but this can be a bit hard to arrange.
  - In our description of ElGamal encryption with modular arithmetic, Bob chose a value a which was a primitive root modulo p. It was not actually necessary to choose a primitive root: the system works essentially as well when a is any value whose order is sufficiently large that computing discrete logarithms to the base a is difficult.
  - Bob can search for such a Q_a by computing $(M!)Q_a$ for a reasonably large value of M and making sure that it is not equal to ∞.
  - Alternatively, Bob could try to find a curve E having a prime number of points on it: then any point other than ∞ will have order N.
  - Bob then chooses a positive integer d that is less than the number of points on E_p (he does not actually need to compute the number of points itself, since he can just choose d to be less than $p - 2\sqrt{p}$) and computes the point $Q_b = dQ_a$.
  - Bob then publishes (E, p, Q_a, Q_b), which serve as his public key.
- Now suppose that Alice wants to send Bob a message $P = (x, y)$.
  - Alice chooses a random integer k less than the number of points on E_p (again, she could simply choose a random integer less than $p - 2\sqrt{p}$) and computes $Q_r = kQ_a$ and $Q_s = kQ_b + P$ on E_p.
  - She then sends the pair (Q_r, Q_s) to Bob.
- If Bob has received a ciphertext pair (Q_r, Q_s), he wishes to recover the value of m.
  - To do this, Bob simply computes $Q_s - dQ_r = (kQ_b + P) - d(kQ_a) = P + kdQ_a - dkQ_a = P$ on E_p.
  - Note of course that Bob would compute the subtraction as $Q_s + d(-Q_r)$, where $-Q_r$ is the additive inverse of Q_r.
- Example: If Bob uses elliptic-curve ElGamal with $E : y^2 = x^3 + 7x + 1$, $p = 44927$, $Q_a = (7772, 14369)$, and $d = 22105$, find Bob's public key, encode the message $P = (14605, 29833)$, and then decode the associated ciphertext.
  - First, Bob computes $Q_b = dQ_a = (39061, 4109)$ using successive doubling. His public key then consists of the quadruple (E, p, Q_a, Q_b).
  - Now, if Alice wants to encode the message P, she chooses a random integer k less than $p - 2\sqrt{p} \approx 44503.08$. Imagine she chooses $k = 23207$.
  - She then computes $Q_r = kQ_a = (30566, 37885)$ and $Q_s = kQ_b + P = (35487, 8262) + P = (40194, 40273)$ and sends them to Bob.
  - Bob receives the ciphertext pair Q_r, Q_s, and then decrypts by evaluating $Q_s - dQ_r = Q_s + (35487, 36665) = (14605, 29833)$, which is indeed the correct plaintext.
  - Remark: For the given parameters, the curve E_p turns out to have a prime number of points (44651) so P necessarily has order 44651 on this curve.
- Like with cryptosystems based on modular arithmetic, the only steps required to implement elliptic curve ElGamal are the point operations on the elliptic curve, which can be done comparatively fast using the successive doubling algorithm. However, it is less obvious why the procedure is secure.
  - Suppose Eve intercepts the transmitted information: she will obtain (E, p) along with Q_a, Q_b, Q_r, and Q_s. She wants to compute $P = Q_s - kQ_b = Q_s - dkQ_a = Q_s - dQ_r$ on E_p.
  - If Eve knows d then she can decrypt using the same procedure Bob uses. However, in order to find d from Bob's public key, Eve would need to determine the value d for which $dQ_a = Q_b$, which is the elliptic curve analogue of computing a discrete logarithm.

- Furthermore, since Alice chooses k randomly, $Q_r = kQ_a$ will essentially be a random point on the curve E_p (technically, it will be a random multiple of Q_a, but this does not tell Eve very much if Q_a has a large order). Likewise, $Q_s = kQ_b + P$ will be essentially random.
- Knowing Q_r alone does not help, because in order to compute k Eve would again need to compute an elliptic-curve discrete logarithm. Knowing Q_s does not help much either, because in order for Eve to compute P she would have to know the value of kQ_b, which in turn would require knowing the value of k.
- Ultimately, like with the modular version of ElGamal, the only obvious method of attack is to compute a discrete logarithm.

- It appears to be much harder to compute elliptic curve discrete logarithms than modular discrete logarithms. Several of the simpler systems have natural analogues:
  - There is a version of the Pohlig-Hellman algorithm that will be effective when the number of points N on E_p has only small prime divisors. (In this case, N plays the role of $p-1$ in the algorithm.) This situation is easy to avoid if the curve E is chosen properly.
  - There is also a version of the baby-step giant-step method whose procedure is essentially identical and requires approximately $p^{1/2}$ steps to compute a discrete logarithm.
  - Here is the algorithm, for completeness: to find a solution to $dQ_a = Q_b$ on an elliptic curve E_p modulo p, choose an integer M such that $M^2 \geq N$, where N is the number of points on E_p. Compute two lists: the points xQ_a for all $0 \leq x \leq N-1$ and the points $Q_b - NyQ_a$ for all $0 \leq y \leq N-1$. Then compare the two lists to find an element that is on both lists: if $xQ_a = Q_b - NyQ_a$, we get a solution $d = x + Ny$.

- However, there does not appear to be any natural analogue of any of the sieving algorithms.
  - The basic reason is that the sieving algorithms all rely on an easily-computed notion of "smallness" of a residue class modulo n that remains consistent under modular multiplication (i.e., the product of two small numbers modulo n remains small modulo n). The idea is then to try to obtain a large number of relations among small primes and use them to compute the discrete logarithms of enough small primes to allow new discrete logarithms to be computed rapidly.
  - However, there is no analogous notion of size that is easy to compute on an elliptic curve modulo p: for one thing, even if the x-coordinate of a point is small, the y-coordinate will look more or less random and very often will be large.
  - Also, even if all the coordinates of particular points are both small, their sum may have very large coordinates due to the modular divisions in the addition law.
  - Finally, even if we were to declare that a point is "small" if it had a small x-coordinate, there is no easy way to see how a large point can be written as a sum of small points that is analogous to the way we can easily factor a big integer that is a product of small primes.

- Since the sieving algorithms do not carry over, and there do not seem to be any other natural algorithms that are comparable, we can achieve a level of security comparable to that of RSA using an elliptic curve cryptosystem with much smaller key sizes.
  - It is estimated, based on the speed of integer factorization algorithms versus the speed of elliptic curve discrete logaritm algorithms, that an elliptic curve cryptosystem with a key size of 256 bits provides security roughly comparable to that of RSA with a key size of approximately 3000 bits.
  - The smaller key size leads to significant savings in computation time, even after accounting for the additional complexity of doing elliptic curve addition versus modular multiplication.
  - In actual practice, since it is a nontrivial problem to count the number of points on a given elliptic curve, many elliptic curve protocols specify using a curve published by an independent authority, such as NIST, that has done the point-counting ahead of time and certifies it as being secure. Of course, this requires a degree of trust that the authority has not intentionally chosen a curve that has some kind of nonobvious "backdoor" (i.e., some clever way of computing discrete logarithms quickly), though in practice it seems unlikely such a backdoor would exist.

- Many implementations use elliptic curves defined over finite fields of characteristic 2, since such fields are particularly amenable to binary arithmetic. Such fields will have 2^n elements for some integer n and have a structure similar to the integers modulo 2^n, except with a different type of multiplication that makes all of the nonzero elements into units.
- We will not delve into the rich and interesting area of finite fields, but we will reiterate that it is necessary to work with a different Weierstrass model over a field of characteristic 2.

5.3.3 Key Exchange and Digital Signatures with Elliptic Curves

- Next, we describe how to create a version of Diffie-Hellman key exchange for elliptic curves using the same techniques.
- First, Alice and Bob jointly choose a large prime p, an elliptic curve E_p modulo p, and a point P on E having large order.
  - Again, as with the construction for ElGamal encryption, there are moderately straightforward procedures for generating this triple (E, p, P).
  - One option is to choose p and then search for an elliptic curve E_p whose number of points is a prime or a prime times another small number d (e.g., 4). Then for any point P, if $dP \neq \infty$, it follows that P must have large order.
  - It is also possible to use a curve certified by a trusted authority that has done the point-counting already, although this involves taking the risk that Eve has precomputed a lot of information about that curve for the purposes of trying to break cryptographic protocols that use it.
  - Alice then chooses a secret positive integer a less than the order of P and sends Bob the point $Q_a = aP$ on E_p.
  - Likewise, Bob chooses a secret positive integer b less than the order of P and sends Alice the point $Q_b = bP$ on E_p.
  - Their secret shared key is then the point $Q_{ab} = abP = a(Q_b) = b(Q_a)$, which can be computed by both Alice and Bob using their secret number along with the point shared by the other.
- Example: Use elliptic-curve Diffie-Hellman to construct a secret shared key using $E : y^2 = x^3 + 7x + 1$, $p = 44927$, and $P = (27844, 29401)$, where Alice's secret number is $a = 40006$ and Bob's secret number is $b = 18846$.
  - Alice computes $Q_a = aP = (3454, 34367)$ and sends it to Bob. Bob computes $Q_b = bP = (22472, 6971)$ and sends it to Alice.
  - Alice then recovers $Q_{ab} = aP_b = (2147, 22480)$ and Bob recovers $Q_{ab} = bQ_a = (2147, 22480)$.
  - Bob and Alice now have a secret shared key $Q_{ab} = (2147, 22480)$ that they can use for further communications (e.g., with a symmetric-key cryptosystem).
- If Eve is eavesdropping on the conversation, she will know E_p along with P, Q_a, and Q_b , and she wants to compute Q_{ab}.
  - In order to do this, Eve would essentially need to compute one of the multipliers a and b. Since P is assumed to have large order, the only reasonable way to do this is for her to evaluate a discrete logarithm on E_p.
  - Again, as we have already discussed, computation of discrete logarithms on elliptic curves appears to be very difficult.
  - It is of course possible that there is some way to combine the information in P, Q_a, Q_b to find Q_{ab}, but this seems unlikely since the operations of scaling a point by a and scaling a point by b are essentially independent.
- Like with modular Diffie-Hellman, we can extend this basic protocol to include more than two participants.

- For example, if Alice, Bob, and Carol wish to construct a secret shared key together, they collectively agree on a triple (E, p, P) and choose their own secret numbers a, b, and c each less than the order of P on E_p.
- Alice then publishes $Q_a = aP$, Bob publishes $Q_b = bP$, and Carol publishes $Q_c = cP$.
- Next, Alice computes $Q_{ab} = aQ_b$ and publishes it, Bob computes $Q_{bc} = bQ_c$ and publishes it, and Carol computes $Q_{ac} = cQ_a$ and publishes it.
- Each person then computes the shared secret key $Q_{abc} = aQ_{bc} = bQ_{ac} = cQ_{ab}$ using their secret and the public information.
- If Eve is eavesdropping, she will have P, aP, bP, cP, abP, acP, bcP, but not the secret $abcP$. There is no obvious way she can compute the secret that does not essentially require computing a discrete logarithm somewhere.

- Also, like with the basic implementation of modular Diffie-Hellman, this protocol does not have any authentication and is therefore susceptible to the "man-in-the-middle" attack wherein Mallory impersonates Alice to Bob and simultaneously impersonates Bob to Alice, and performs a simultaneous key exchange with both of them.
  - One way to include an authentication step would be for both of Alice and Bob to put a digital signature on their communications during the key creation process, so that the other person feels confident that Mallory is not impersonating either of them.

- We will now describe how to adapt the ElGamal signature algorithm to the elliptic curve setting. Some details of the algorithm differ from the modular case since we are dealing with points rather than individual numbers.
  - Alice first creates an elliptic-curve ElGamal public key (p, E, Q_a, Q_b) where p is a large prime, E is an elliptic curve modulo p on which it is hard to compute discrete logarithms, Q_a is a point on E whose order has only large prime factors, and $Q_b = dQ_a$ for Alice's secret number d.
  - Alice also calculates the number of points N on E_p.
  - If Alice now wants to sign a message m, which we consider to be an integer modulo N, she first chooses a random positive integer k relatively prime to N.
  - Alice then computes $Q_r = kQ_a = (x, y)$ and $s = k^{-1}(m - dx) \pmod N$, and sends Bob her signed message (m, Q_r, s).
  - Bob verifies that Alice's signature is correct by computing $xQ_b + sQ_r$ and comparing it to mQ_a. If the results are equal, he accepts the signature, and otherwise he rejects it.
  - The verification works because $xQ_b + sQ_r = x(dQ_a) + s(kQ_a) = (m - dx)Q_a = xdQ_a + mQ_a - dxQ_a = mQ_a$, where we are using the fact that $sk \equiv m - dx \pmod N$ to deduce that $ksQ_a = (m - dx)Q_a$ since the order of Q_a necessarily divides N.

- As with the elliptic-curve ElGamal encryption scheme, the security of this procedure ultimately relies on the difficulty of computing a discrete logarithm and the fact that k is randomly chosen.
  - It does not depend on the difficulty of computing the number of points on the curve N, which could even be published as part of the public key if desired.

- Example: Alice publishes her elliptic-curve ElGamal signature key with $E : y^2 = x^3 + 7x + 1$, $p = 44927$, $Q_a = (3174, 1067)$, and $Q_b = dQ_a = (38921, 25436)$ with her secret $d = 25661$. Bob then sends her the message $m = 17781$. Generate a signature for this message with $k = 33050$ and verify that it is correct.
  - Alice computes the number of points on the curve, $N = 44651$, which happens to be prime.
  - She then computes $Q_r = kQ_a = (11123, 34794) = (x, y)$ and $s = k^{-1}(m - dx) \equiv 42665 \pmod N$.
  - She then sends the pair (Q_r, s) to Bob, who then evaluates $xQ_b + sQ_r = (29063, 26534) + (36219, 42811) = (35670, 7590)$ and compares it to $mQ_a = (35670, 7590)$.
  - The results are equal, so Bob accepts the signature.

Well, you're at the end of my handout. Hope it was helpful.