A Brief Look at Elliptic Curve Cryptography

It turns out that RSA requires more bits, with accompanying computational time and memory space, than Elliptic Curve Cryptography (ECC) to achieve the same security level. For this reason, we will take a quick look at ECC. Plus, it uses Z_n and divide-and-conquer.

Elliptic Curves in the Plane

To motivate ECC, we need to briefly consider elliptic curves in the ordinary real number plane.

Consider the *elliptic curve* in the familiar (say from calculus) real number plane defined by

$$y^2 = x^3 + ax + b,$$

where a and b are any real numbers (except for mysterious reasons to be sort of explained later, we require $4a^3 + 27b^2 \neq 0$).

Note that for any x, we can compute $x^3 + ax + b$, and if it turns out to be positive, then there are two values of y such that $y^2 = x^3 + ax + b$, namely

$$y = \pm\sqrt{x^3 + ax + b}.$$

And, of course, if $x^3 + ax + b$ turns out to be negative, there are no points on the curve with that x coordinate. If $x^3 + ax + b = 0$, then there is one point on the curve with that x coordinate.

> If we go to `wolframalpha.com` and type in
> `plot elliptic curve y^2 = x^3-3x+7` (or use whatever a and b we wish), we can get a graph of an elliptic curve.
>
> It turns out that if $4a^3 + 27b^2 > 0$, then the curve has one connected piece, and if $4a^3 + 27b^2 < 0$, then it has two.

It turns out to be a very interesting question to ask "how do lines intersect an elliptic curve?" Looking at some graphs, we can see that vertical lines typically hit an elliptic curve in two points, with opposite y values (or miss it entirely, or hit a single special point). If we pick any two points on the elliptic curve with different x values and draw the line through them, it appears that this line hits the curve in one other point. Soon we will suggest why this is true, and use it in ECC.

How Lines Hit an Elliptic Curve

First, a vertical line clearly either misses the elliptic curve entirely, or hits it in two points with the same x value and y values that are opposites of each other.

For the more interesting cases, consider a non-vertical line with slope λ that passes through a point (x_P, y_P) on an elliptic curve $y^2 = x^3 + ax + b$.

The equation of the line is

$$y = y_P + \lambda(x - x_P).$$

Substituting into the elliptic curve, we have

$$(y_P + \lambda(x - x_P))^2 = x^3 + ax + b,$$

or

$$y_P^2 + 2\lambda y_P(x - x_P) + \lambda^2(x^2 - 2xx_P + x_P^2) = x^3 + ax + b.$$

Simplifying a little and getting 0 on the right-hand side, we have

$$x^3 + ax + b - y_P^2 - 2\lambda y_P(x - x_P) - \lambda^2(x^2 - 2xx_P + x_P^2) = 0.$$

Now, we might expect to have to do a bunch of algebra with this equation and hope to somehow eventually get formulas for the three roots x, but we can be much more clever.

First, it is easy to see that the coefficient of x^2 in this equation is $-\lambda^2$.

Second, the Fundamental Theorem of Algebra says that any polynomial of degree n has n (possibly complex and possibly not distinct) roots, and the polynomial can be expressed as a product of all the terms like $x - a$ where a is a root. In our simple case, since the coefficient of x^3 in the polynomial is 1, the equation has to be

$$(x - x_P)(x - z_1)(x - z_2) = 0,$$

where x_P is the root of the equation we know about, and z_1 and z_2 are the other two roots. From another fact of abstract algebra, we know that either z_1 and z_2 both have imaginary parts (and are actually conjugates of each other—i.e., $z_1 = c + di$ and $z_2 = c - di$), or are both real.

Let's assume that the line hits the elliptic curve in at least two points, so all three roots are real. In this case, we can denote the roots by $z_1 = x_Q$ and $z_2 = x_S$, letting Q and R be the other (besides P) two points where the line hits the elliptic curve.

Note, though, that we don't know that x_P, x_Q, and x_R are distinct.

If we multiply out $(x - x_P)(x - x_Q)(x - X_R)$, the x^2 coefficient turns out to be $-(x_P + x_Q + X_R)$, thus

$$\lambda^2 = x_P + x_Q + x_R.$$

or

$$x_R = \lambda^2 - x_P - x_Q.$$

Then by using the equation of the line, we obtain

$$y_R = y_P + \lambda(x_R - x_P),$$

so we have formulas for the third point R if the line hits at points P and Q.

We can now handle all cases that we care about.

First, suppose that for some reason we have two distinct points P and Q that we care about on an elliptic curve. Then we can compute the slope of the secant line through these points, namely

$$\lambda = \frac{y_P - y_Q}{x_P - x_Q}.$$

Then the previous discussion gives us the formula for $R = (x_R, y_R)$ of a third distinct point on the line and elliptic curve.

The other interesting situation is if we have a single point P on an elliptic curve, and we want to find where the tangent line to the curve at P hits the curve.

If we take the derivative with respect to x of both sides of the elliptic curve equation we obtain

$$2yy' = 3x^2 + a,$$

or

$$y' = \frac{3x^2 + a}{2y}.$$

In this case, we have

$$\lambda = \frac{3x_P^2 + a}{2y_P},$$

and we can use the formulas for the components of R to find another point where the tangent line hits the curve.

> Note that if we take a point Q on the elliptic curve close to P, from the previous section we see that the line through P and Q will hit the elliptic curve at some other point R. But, in the limit as Q approaches P, this line will become the tangent line at P. So, essentially the two cases are the same, where in the first case $P \neq Q$, and in the second $P = Q$.

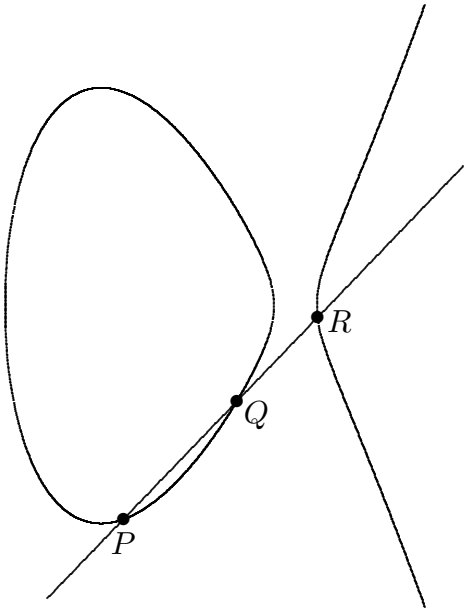
Examples of Line-Curve Intersections

Here are pictures of three important types of line-elliptic curve intersections, all for the elliptic curve

$$y^2 = x^3 - 5x + 4.$$

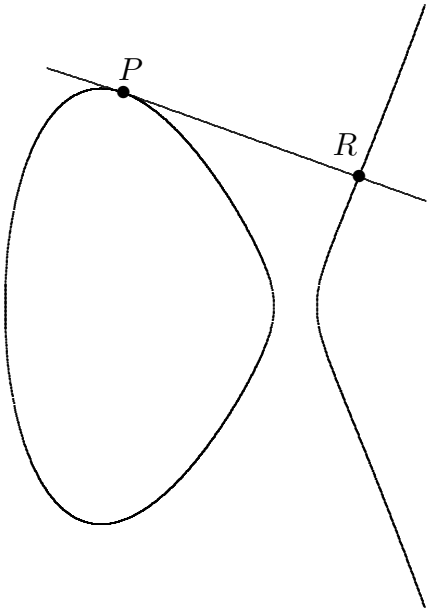
Secant Case

Here is the line through distinct points P and Q hitting a third point R:



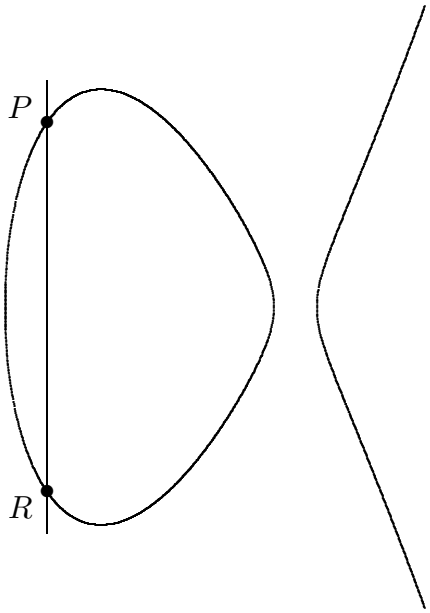
Tangent Case:

Here is a picture of the tangent line at a single point P hitting the elliptic curve at the point R:



Vertical Case:

Here is a picture of the vertical line through P hitting the point R with the same x coordinate:



Solving a Cubic with no Quadratic Term

Note: this section can be safely ignored, but if you are really into algebra, ...

To see where the mysterious "$4a^3 + 27b^2 \neq 0$" thing comes from, and maybe why elliptic curves are defined without an x^2 term, we now show that we can solve

$$x^3 + px + q = 0$$

easily, as follows.

Let

$$x = w - \frac{p}{3w}.$$

Then the equation becomes

$$(w - \frac{p}{3w})^3 + p(w - \frac{p}{3w}) + q = 0.$$

Multiplying out everything, and simplifying a little, we obtain

$$w^3 - 3w^2\frac{p}{3w} + 3w\frac{p^2}{9w^2} - \frac{p^3}{27w^3} + pw - \frac{p^2}{3w} + q = 0,$$

or

$$w^3 - pw + \frac{p^2}{3w} - \frac{p^3}{27w^3} + pw - \frac{p^2}{3w} + q = 0,$$

which is lovely and sort of shows why this substitution was made, because this simplifies to

$$w^3 - \frac{p^3}{27w^3} + q = 0,$$

which after multiplying through by w^3, gives

$$w^6 + qw^3 - \frac{p^3}{27} = 0.$$

This is a quadratic in w^3, so

$$w^3 = \frac{-q \pm \sqrt{q^2 + 4\frac{p^3}{27}}}{2} = -\frac{q}{2} \pm \sqrt{\left(\frac{q}{2}\right)^2 + \left(\frac{p}{3}\right)^3}.$$

> This equation gives two values for w^3, each of which will give 3 roots, for a total of 6 roots, as expected for a sixth-degree polynomial, but after substituting these 6 values for w to get x, only three of these will work in the original cubic equation.

This suggests why the elliptic curve with $b^2 + \frac{4}{27}a^3 = 0$ (or $4a^3 + 27b^2 = 0$) is ruled out, because this is the case where this cubic has duplicate roots.

Elliptic Curves over Z_p

The previous discussion of elliptic curves over the real numbers was really just motivation for doing elliptic curves over Z_p, as we will now describe.

Let p be a prime number, so Z_p is a field. In the following, all arithmetic calculations are done in Z_p.

> Don't confuse p with P.

We now define a *group* (an algebraic structure with one operation that is associative, has an identity element, and for which every group member has an inverse for the operation) as follows. This group is inspired by elliptic curves in the real plane, and in particular by intersections of lines with such curves.

Choose any a and b in Z_p (with $4a^3 + 27b^2 \neq 0$), and use the elliptic curve

$$y^2 = x^3 + ax + b.$$

As the elements of the group, take all points (x, y) where x and y are in Z_p and (x, y) is on the elliptic curve, along with a special "point at infinity" denoted by 0.

The operation of the group will be denoted by $+$, and 0 is officially declared to be the identity element for the operation, so for all P in the group (include 0),

$$P + 0 = P.$$

Inspired by the properties of how lines intersect elliptic curves over real numbers, we want to say that for any three group members P, Q, and R that lie on a line,

$$P + Q + R = 0.$$

> Note that this definition of + doesn't say anything about the order in which the points occur, so straight from the definition we have the fact that the operation is *commutative*.

This one idea lets us define addition of any two group elements, and the inverse of every group element, as follows.

The easiest case is where P and Q are distinct points, and R is the third point on the line and curve. This case was named the "secant case" in our earlier discussion. Note that really this case is totally symmetrical—we can take any two distinct group members on a line and uniquely find the third. In this case, we can use two points to determine the secant line slope λ, and then compute the third point.

The secant case works for $P \neq Q$. What if we want to compute $P + P$? This is the "tangent case," where we use the slope of the tangent line at P to give us a reasonable slope and use the formulas to compute R. In this case we are essentially saying

$$P + P + R = 0,$$

because the tangent line is the limit of the secant lines, with corresponding R, obtained by taking points Q closer and closer to P.

The final case is a vertical line through $P = (x_P, y_P)$, which hits the elliptic curve at $R = (x_P, -y_P)$. This case explains a lot. First, we define the inverse of P as this R, and write

$$-P = (x_P, -y_P).$$

This case fits our general definition if we think of the special group member 0 as "the point at infinity" and imagine that all vertical lines intersect at 0, so 0 is on every vertical line, so it makes sense to say

$$P + (-P) + 0 = 0.$$

Finally, we note that simple group algebra gives us our actual definition of $P + Q$ in each case, resulting in

$$P + Q = -R$$

in the secant case (using the secant slope), and

$$P + P = -R$$

in the tangent case (using the tangent slope).

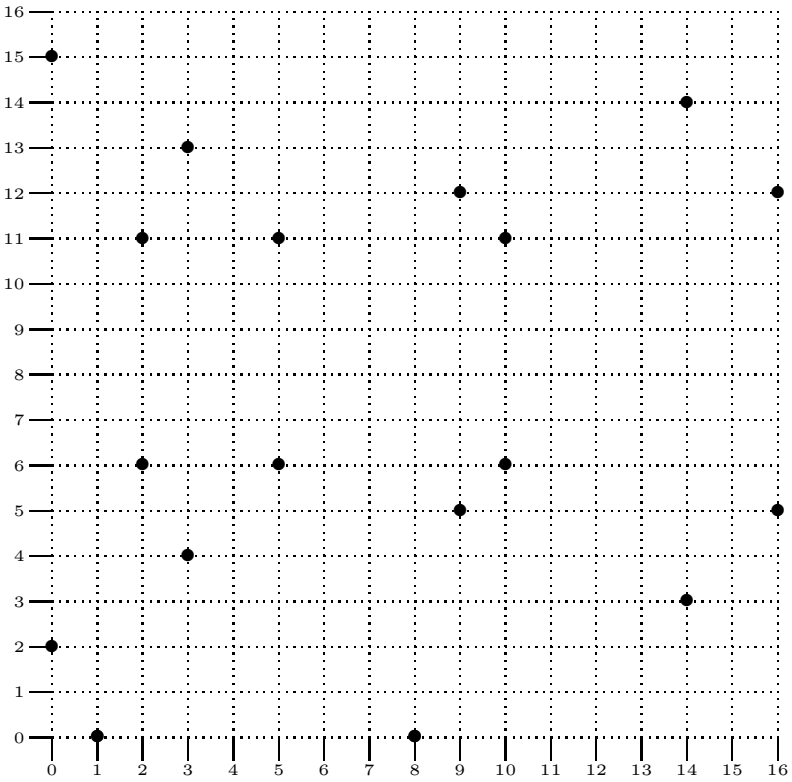
Examples of Performing Operations in the ECC Group

Let's use the same elliptic curve we did in the real number case:

$$y^2 = x^3 - 5x + 4$$

and let's use $p = 17$. Note that in Z_{17}, $a = -5 = 12$.

Here is a picture of all the points in the resulting group:



Let's start with the base point $P = (3, 4)$ and do some group additions.

First, let's do $2P = P + P$. This is the tangent line case, so we first compute

$$\lambda = \frac{3x_P^2 + a}{2y_P} = \frac{3 \cdot 9 + 12}{2 \cdot 4} = \frac{5}{8}.$$

In Z_{17}, $\frac{1}{8} = 15$ since $8 \cdot 15 = 120 = 1 + 119 = 1$. So, $\lambda = 5 \cdot 15 = 7$.

Now we simply compute

$$x_R = \lambda^2 - x_P - x_P = 7^2 - 3 - 3 = 15 - 6 = 9,$$

and

$$y_R = y_P + \lambda(x_R - x_P) = 4 + 7(9 - 3) = 4 + 7 \cdot 6 = 12.$$

Finally, we obtain $2P = -R = (9, 5)$, since $-12 = 5$.

Now let's compute $3P = 2P + P = (9, 5) + (3, 4)$:

This is the secant line case so we first compute

$$\lambda = \frac{4-5}{3-9} = \frac{16}{11} = 16 \cdot 14 = 3$$

(skillfully figuring additive and multiplicative inverses of numbers in Z_{17}).

Then we compute

$$x_R = \lambda^2 - x_P - x_Q = 3^2 - 9 - 3 = -3 = 14,$$

and

$$y_R = y_P + \lambda(x_R - x_P) = 5 + 3(14 - 9) = 3$$

so $R = (14, 3)$, and $3P = S = (14, 14)$.

Continuing in this way, we eventually reach $17P = (3, 13)$. Let's compute $18P = 17P+P = (3, 13) + (3, 4)$.

This looks like a secant case—the two points are distinct—so we compute

$$\lambda = \frac{4-13}{3-3} = \frac{4}{0} = ???$$

At this point we realize that $(3, 13)$ and $(3, 4)$ are inverses, so $18P = 0$.

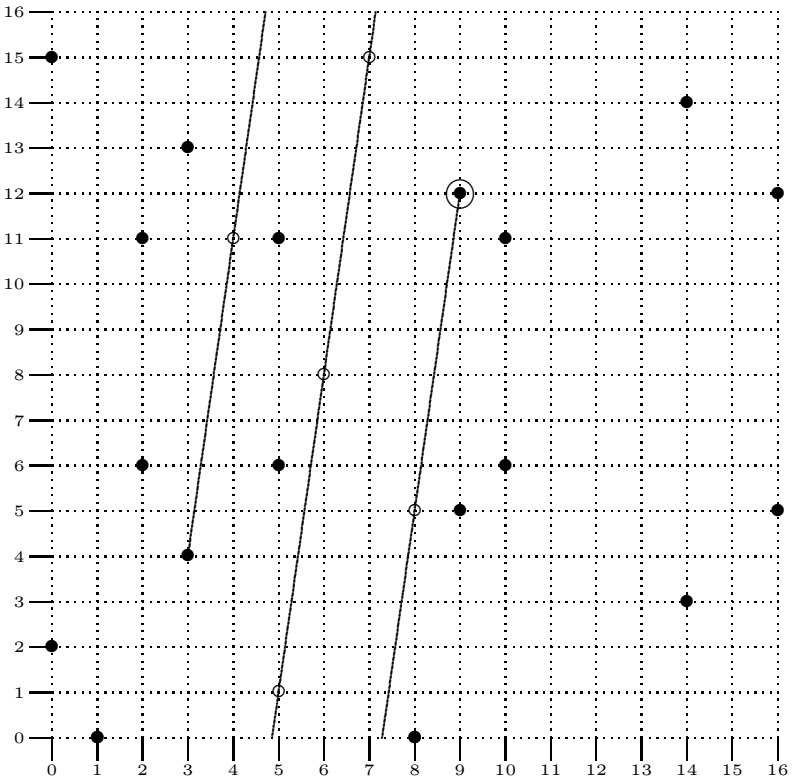
We can also observe that this is of course a vertical case, since $(3, 13)$ and $(3, 4)$ are on the same vertical line, so their sum must be the point at infinity.

Sanity Check

At this point we might be worried that we are using formulas from the real number realm to do things in the Z_p world, which might be wrong. The first thing we can say is that most of what we did in the real number realm was algebraic, not geometric, and all we really used were field properties.

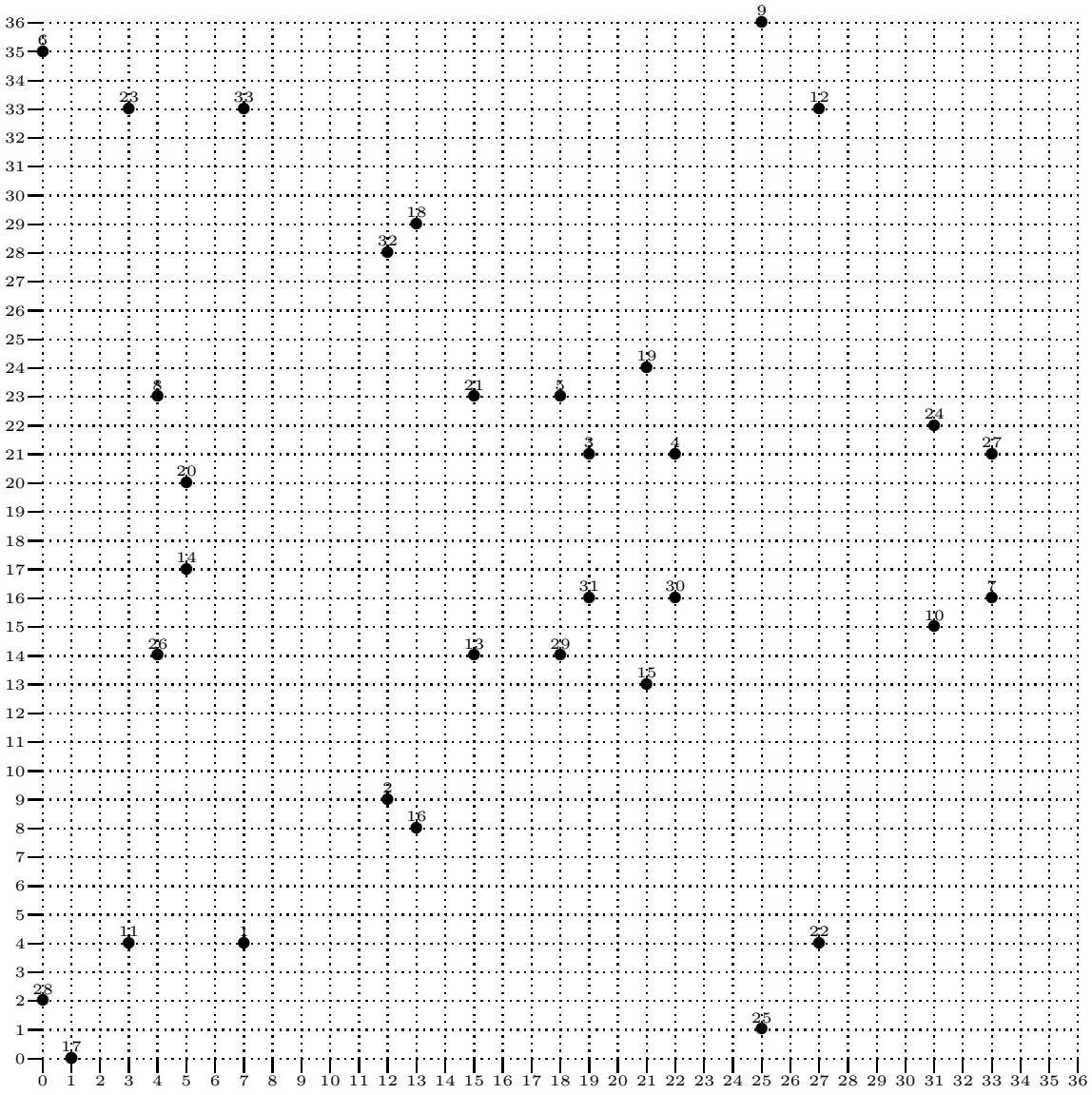
But, we also used "lines" in a big way, so we should check that these line ideas make sense in the Z_p world (because, for example, comparisons of the sizes of numbers totally do not make sense in Z_p, so maybe lines don't either).

To somewhat address this concern, note in the picture below that we have headed away from $P = (3, 4)$ with a slope of $\lambda = 7$, as we calculated from the real number-based formula for the slope of a tangent line. Interpreting slope 7 to mean that we move one unit to the right for every 7 units up, and carefully noting that when we go one unit up from row 16 we move to row 0, we see that things do work out—the line is wrapped and does eventually hit $S = (9, 12)$, just as we calculated.


16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Some Pictures

Mostly because I wrote the code to produce this diagram and want to use it, but also to suggest how randomly chaotic the sequence of points mP is, here is a picture of all the points on the elliptic curve $y^2 = x^3 - 5x + 4$, over Z_{37}, with base point $(7, 4)$, with the points labeled by their scalar multiple of $(7, 4)$:

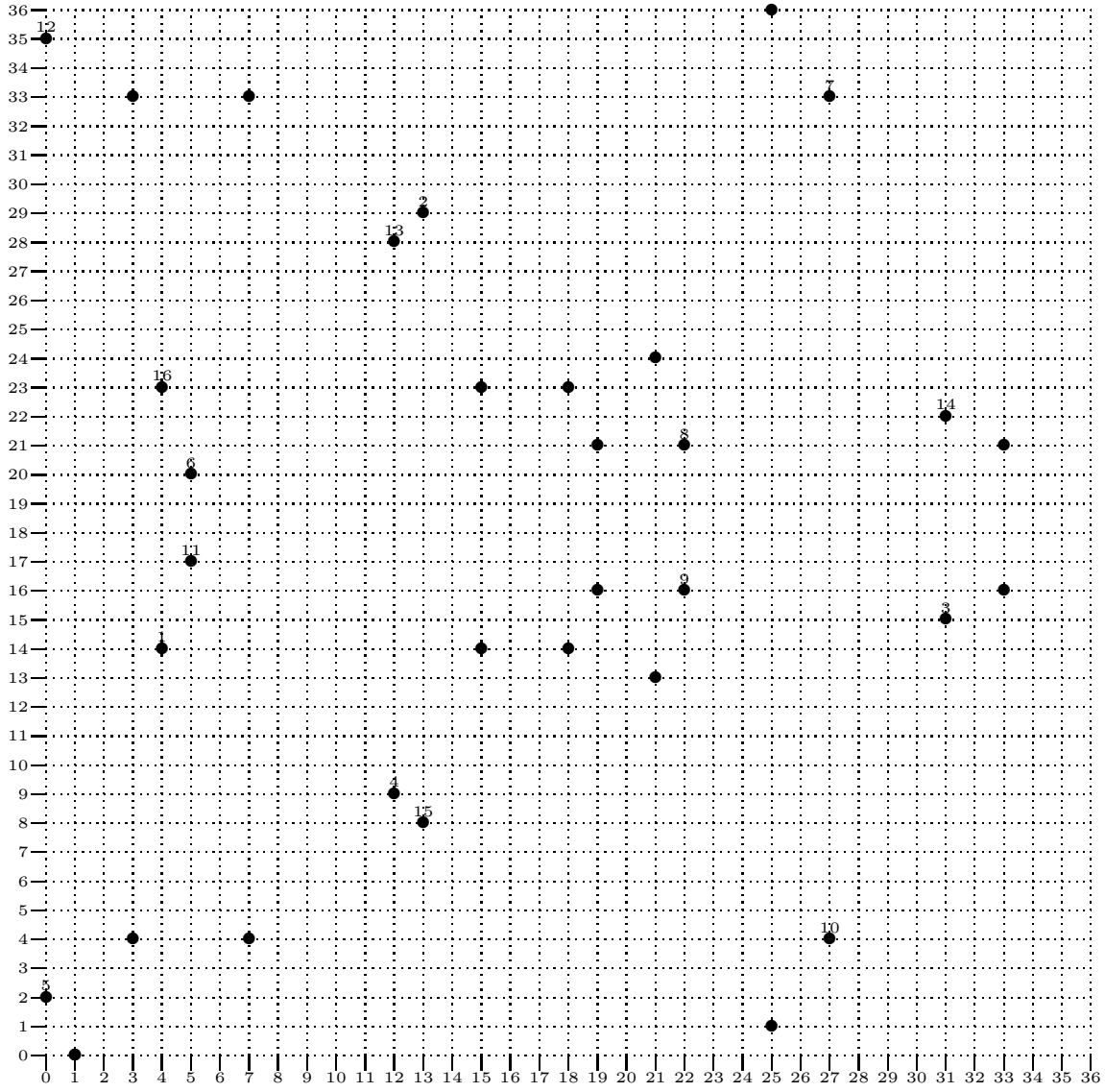


Note that all the points on the elliptic curve over Z_{37} are hit, reaching $33(7, 4) = (7, 33)$, and then reaching the point at infinity, 0, because

$$(7, 33) + (7, 4) = 0$$

since $(7, 33)$ and $(7, 4)$ are opposites in the group.

It turns out $(7, 4)$ was a good base point, because for other starting points, not all points on the curve are hit. For example, here is the similar diagram, starting at $(4, 14)$:



Computing any Multiple of a Point Efficiently

Given our usual situation, namely an elliptic curve $y^2 = x^3 + ax + b$, a prime number p, a base point P, and a positive integer m, we define

$$mP = \overbrace{P + P + \cdots + P}^{m}.$$

The previous discussion explained how to compute

$$2P = P + P$$

by using the tangent line slope for λ, and how to compute

$$P + Q$$

for $P \neq Q$ by using the secant line slope for λ.

Now we just need to realize that mP can be computed efficiently by using exactly the same idea as for modular exponentiation in the RSA stuff, except that our group operation for ECC is addition instead of multiplication, so mP is analogous to a^e in RSA.

In case that's not obvious enough, here is an example of this idea, where $m = 25$:

We simple repeatedly double (analogous to repeatedly squaring), computing these values:

Repeated Doublings	Repeated Halvings	Accumulated Answer	
P	25	P	(25 is odd, so add P)
$2P$	12	P	(12 is even, so don't add $2P$)
$4P$	6	P	(6 even so don't add $4P$)
$8P$	3	$P + 8P$	(3 odd so add $8P$)
$16P$	1	$P + 8P + 16P = 25P$	(1 odd so add $16P$)

Note that the repeated doubling part of the algorithm involves the tangent form of adding a point to itself, while the accumulating part involves the secant form of adding two points that aren't equal.

⇒ Project 18 [routine] Computing mP efficiently

Demonstrate the efficient algorithm for computing $19P$ using elliptic curve $y^2 = x^3+2x+3$, over Z_{17}, and base point $P = (2, 7)$.

> You should do this Project by hand/calculator, but note that the `Code` folder at the course web site contains a folder `ECC` that contains some classes that could be used to do this computation.

For your convenience, here are the Z_{17} multiplication table and a bunch of multiples of 17:

·	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
2	0	2	4	6	8	10	12	14	16	1	3	5	7	9	11	13	15
3	0	3	6	9	12	15	1	4	7	10	13	16	2	5	8	11	14
4	0	4	8	12	16	3	7	11	15	2	6	10	14	1	5	9	13
5	0	5	10	15	3	8	13	1	6	11	16	4	9	14	2	7	12
6	0	6	12	1	7	13	2	8	14	3	9	15	4	10	16	5	11
7	0	7	14	4	11	1	8	15	5	12	2	9	16	6	13	3	10
8	0	8	16	7	15	6	14	5	13	4	12	3	11	2	10	1	9
9	0	9	1	10	2	11	3	12	4	13	5	14	6	15	7	16	8
10	0	10	3	13	6	16	9	2	12	5	15	8	1	11	4	14	7
11	0	11	5	16	10	4	15	9	3	14	8	2	13	7	1	12	6
12	0	12	7	2	14	9	4	16	11	6	1	13	8	3	15	10	5
13	0	13	9	5	1	14	10	6	2	15	11	7	3	16	12	8	4
14	0	14	11	8	5	2	16	13	10	7	4	1	15	12	9	6	3
15	0	15	13	11	9	7	5	3	1	16	14	12	10	8	6	4	2
16	0	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1

$0, 17, 34, 51, 68,$

$85, 102, 119, 136, 153,$

$170, 187, 204, 221, 234,$

$255, 272, 289, 306, 323, 340$

The Discrete Logarithm Problem

Now we want to consider how the previous material applies to cryptography.

> Recall that in RSA the "one way" process was multiplying together two large prime numbers p and q, obtaining n which was published. The security of RSA is based entirely on the difficulty of reversing this, namely finding p and q when given n.

The one way process for ECC is known as the *discrete logarithm problem*. Given a large positive integer m and a base point P, the previous algorithm shows that we can compute mP easily. But, given a point Q in the ECC group, it is very difficult to find m such that $Q = mP$.

> The name "discrete logarithm problem" comes from the problem of finding, given b, m such that $a^m = b$ in Z_p. In ECC, since we use + instead of $\cdot$, the notation given is analogous.

Diffie-Hellman-Merkle Key Exchange

Diffie and Hellman (with credit given by Hellman to Merkle when he realized everyone was naming the process "Diffie-Hellman," even though Merkle had the original idea) figured out a way that two entities, say named Alice and Bob, could generate a *shared secret*. We will use ECC for the details of their process.

First, somebody publishes a, b, p, and P, specifying use of the elliptic curve $y^2 = x^3+ax+b$ over Z_p, with base point P.

Alice has a secret number, say m. She efficiently computes mP and publishes this result as the point A.

Bob has a secret number n, efficiently computes nP, and publishes this result as the point B.

Alice takes B and efficiently computes mB, using her knowledge of m.

Bob takes A and efficiently computes nA, using his knowledge of n.

Now, even though everyone can see a, b, p, P, A, and B, only Alice and Bob know the point mnP, which is their shared secret. Alice knows this point because

$$mB = m(nP) = mnP,$$

and Bob knows this point because

$$nA = n(mP) = nmP = mnP.$$

> This is not a course in computer security, so we won't pursue how Alice and Bob can use their shared secret of these two integers—the x and y coordinates of mnP, which are each in Z_p, except to point out that these numbers could be used as a code book. Alice could take a message and encode it using `Encode`, producing a sequence of pairs of digits, and then use the digits of mnP somehow to encrypt those digits, in a way that she and Bob had publicly discussed, and then Bob could use the digits of mnP to decrypt those digits, and use `Decode` to get the original plain-text message.

⇒ Project 19 [routine] Breaking Tiny ECC

Suppose Alice and Bob have agreed to use $y^2 = x^3 - 17x + 31$, $p = 52981$, and $P = (107, 391)$. They exchange their public keys mP and nP, but your agents sadly are unable to see those points. Later Alice sends the encrypted message 77066213 to Bob, using their shared secret to encrypt her original plain-text message. Bob gets sloppy and leaves a piece of paper in plain sight on his desk, which your secret agent sees, reporting to you that Bob wrote down `fl` followed by two more symbols that couldn't be read.

You know all this information, and suspect that Alice and Bob are using ECC, are using `Encode` and `Decode`, and are using their shared secret (which is mnP) as follows: take

the two least significant digits of x and add them to the highest two digits of the encoded message, reducing mod 100. Then use the next 2 digits of x to encrypt the next two digits of the encoded message. Then use the two least significant digits of y to encode the next two digits of the encoded message, and then the next two digits of y to encode the last two digits of the encoded message. When Bob receives the encrypted message, he will decode the pairs of digits in the same way, but will subtract the digits of the shared secret number mod 100.

For example, suppose the shared secret number were $(1296, 35142)$. To send the message `help`, Alice would first use `Encode` to get 73707781. The encrypt/decrypt digits are 96, 12, 42, and 51, so Alice would compute

$$73 + 96 = 169 = 69,$$

$$70 + 12 = 82,$$

$$77 + 42 = 119 = 19,$$

$$81 + 51 = 132 = 32,$$

so she would send the message 69821932.

Bob would then subtract the corresponding pairs of digits of their shared secret, obtaining

$$69 - 96 = -27 = 73,$$

$$82 - 12 = 70,$$

$$19 - 42 = -23 = 77,$$

$$32 - 51 = -19 = 81$$

for a decrypted message 73707781, which he would then `Decode`, giving `help`.

Your job on this Project is to first figure out m and n, use them to find mnP, and then figure out what message Alice sent.

Hint: use the `PointECC` class.

⇒ Project 20 [optional] Breaking ECC by Social Hacking

Suppose Alice and Bob have agreed to use these values for exchanging encrypted information (following the published `secp256k1` standard):

```
p = 115 792 089 237 316 195 423 570 985 008 687 907 853
    269 984 665 640 564 039 457 584 007 908 834 671 663
a = 0
b = 7
P = ( 55 066 263 022 277 343 669 578 718 895 168 534 326
     250 603 453 777 594 175 500 187 360 389 116 729 240
      ,
      32 670 510 020 758 816 978 083 085 130 507 043 184
     471 273 380 659 243 275 938 904 335 757 337 482 424 )
```

Alice is overheard in a restaurant somewhat pompously explaining that she has this great technique for making the computation of mP easier—she just uses for her secret number m a power of 2 plus a two-digit number, which makes the efficient algorithm even more efficient, because most of the work is doubling repeatedly to get up to the desired power of 2, with a little extra work to add on the small extra multiple of P.

Alice sends to Bob, on an insecure communication channel as usual, her value for mP, namely

```
(  98 333 898 174 860 222 763 621 164 809 560 426 900
  902 581 988 820 015 661 720 799 616 398 614 033 468
,
   60 703 462 459 530 085 880 474 331 476 429 053 299
  167 650 471 903 125 187 953 999 331 805 378 093 068 )
```

Your job on this Project (and please work alone on this Project, and do not share your answer with anyone) is to determine Alice's secret number m.

Submit your work by email, stating Alice's secret number m, attaching any code that you wrote/modified to get this value, and a brief description of your reasoning and process.