# Cryptography Homework 7: Elliptic Curve Diffie-Hellman Key Exchange (ECDH)

Do this lab in groups of two (or three if there is an odd number of students.)

Note: The series of four blogs on the subject, [Elliptic Curve Cryptography: a gentle introduction](http://andrea.corbellini.name/2015/05/17/elliptic-curve-cryptography-a-gentle-introduction), by Andrea Corbillini, is awesome! We covered some of her basic material from the first blog in class. The remaining blogs cover finite fields (subgroups and base points, very important), details of ECDH and ECDSA (EC Digital Signature Algorithm), and attacks against discrete logarithm problems. If you are at all interested in ECC, her blogs are the place to start.

## A simple key exchange

Use Andrea’s [calculator for EC multiplication over a finite field](https://cdn.rawgit.com/andreacorbellini/ecc/920b29a/interactive/modk-mul.html) and select the curve y2 = x3 + 5x + 3 with a modulus, p, of 97. Together, select a point that will be the base point, P. The curve, modulus, and base point will be public knowledge. Record them.

Individually, each partner should select a number for n that is greater than 1 and less than 91 to be their private key. Use the calculator to determine the coordinates of their public key (Q = n\*p in the calculator. (The entire public key is the curve, p, P, and the coordinates Q that you computed.)

Exchange public keys with your partner. Compute the shared key by putting your partner’s public coordinates into P, and your private number into n. The value of Q in the calculator is your shared session key and should be the same as the value your partner gets.

Our modulus is 97. How many bits of security do we have (i.e., how many bits does it take to represent 97)? Our shared key x coordinate can be represented by that many bits, as can our y coordinate. Can we double the security of our key by concatenating x and y to double the number of bits? Why or why not? (Hand in 1)

Note the lines on the calculator below the coordinates for Q: “The curve has 91 points (including the point at infinity). The subgroup generated by P has 91 points.” That will be useful in the next part.

## Subgroups

When cryptographers select a curve and modulus, they like the number of points on the curve to be a prime number, or at least have large prime numbers as factors. If the number of points is factorable, the field we are using will contain subgroups. Each subgroup will contain the same number of points as the factors of the number. For example, if the number of points is 30, there will be subgroups with 2, 3, 5, 6, 10, 15, and 30. If the base point you select is in a small subgroup, your security is weakened.

Set your calculator to the curve y2 = x3 + x + 2, with modulus p = 97. How many points are on the curve? What are the factors of that number? (Hand in 2) Experiment with the following base points:  
(1, 2)  
(4, 19)  
(15,26)  
(26,12)  
Which one provides the best security? Which one provides the worst security? (Hand in 3)

The problem of subgroups is common to all encryption that uses finite fields (i.e., most of them), not just ECC.

## Choose your own

With your partner, choose a curve, modulus, and base point that gives you a group or subgroup with more than 200 points. If the group has a prime number of points you don’t have to be so careful about your selection of a base point.

# Hand in

1. Does using both the X and Y coordinates of the shared key increase security? Why or why not?
2. How many points are on the curve y2 = x3 + x + 2, with modulus p = 97? What are the factors of that number?
3. Which point provides the best security?
4. From the Choose your own paragraph, submit the curve, modulus, base point, and number of points in the subgroup.