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

There is an example of what you need to do for this lab in ‘Cryptography Homework 7 example.docx’

Elliptic Curve vs. straight Diffie-Hellman

Diffie-Hellman is based on taking large numbers to large powers. In modular arithmetic it is difficult to discover N, or determine how many times you need to multiply B times itself to get A.

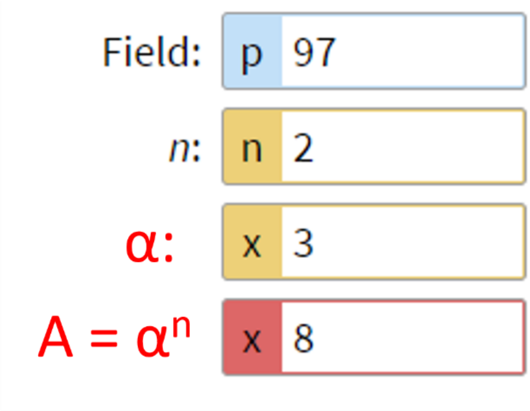
$$A = \alpha^N = \underbrace{\alpha*\alpha*\alpha*\alpha*\alpha*\alpha*\ldots\ \alpha}_{\text{Multiply N times}}$$

All we needed to compute α^N `mod Field` (In RSA we used N for the modulus but I have already used N, and the calculator we will use calls it Field) was some simple Python statement.

```
A = pow(α, N, Field)
```

Since the calculator we will use for elliptic curves is more complicated, we will create a pretend calculator for Diffie-Hellman. That will make it easier to relate what we did with Diffie-Hellman to what we will do with elliptic curves. The calculator below is doing the same thing as this Python statement.

```
8 = pow(3, 2, 97)
```

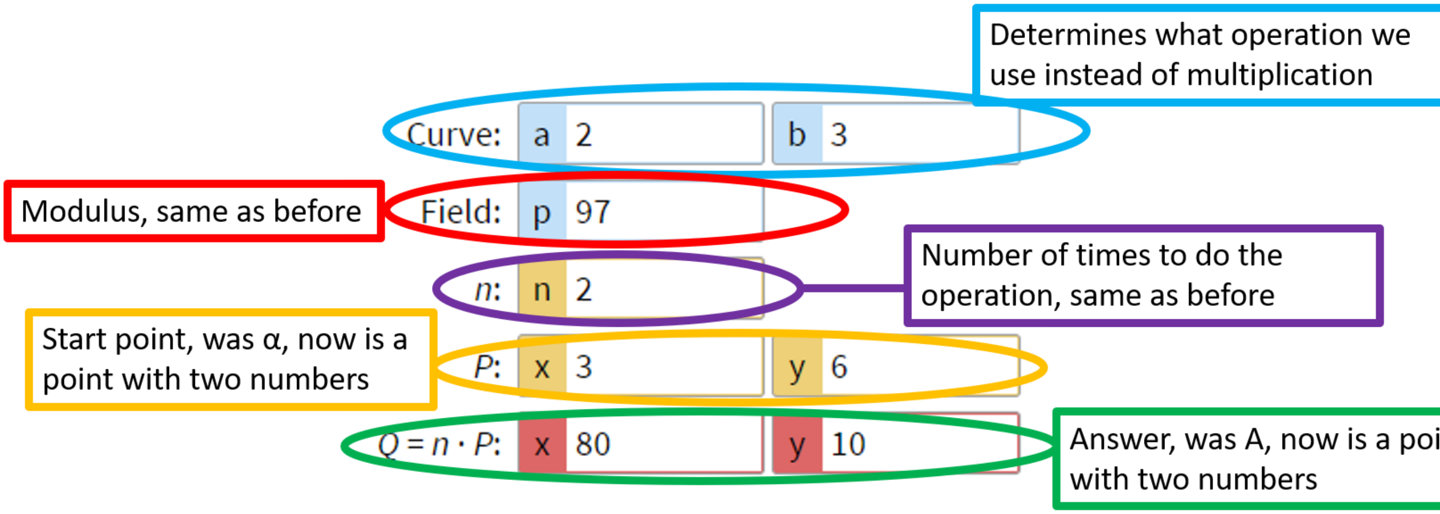


Field:	p	97
n:	n	2
α:	x	3
$A = \alpha^n$	x	8

The differences between straight Diffie-Hellman and elliptic curve (ECDH) are:

- ECDH uses a point P(x, y) with two numbers instead of a single number α
- The answer in ECDH is a point Q(x, y) with two numbers instead of a single number A
- ECDH replaces multiplication with an operation on an elliptic curve

The elliptic curve calculator looks different from the one we just invented for straight Diffie-Hellman. The Curve selection at the top picks the equation we will use, which determines how we will replace multiplication. The other difference is that α and A are now points with two numbers each.



Determines what operation we use instead of multiplication

Curve: a 2 b 3

Modulus, same as before

Field: p 97

n: n 2

Number of times to do the operation, same as before

Start point, was α, now is a point with two numbers

P: x 3 y 6

$Q = n \cdot P$: x 80 y 10

Answer, was A, now is a poi with two numbers

Lab

Do this lab in groups of two (or three if there is an odd number of students.) If you are doing this online, you just have to be both sides of the key exchange.

Note: The series of four blogs on the subject, 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, 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.

Also, this is a particularly good explanation of DHCE with less math.
https://www.youtube.com/watch?v=NF1pwjL9-DE&lc=UgwQZ8eXPqNSmlzHcAp4AaABAg

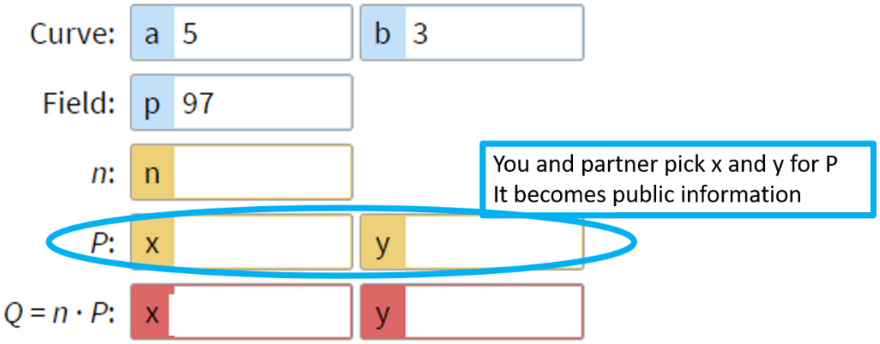
A simple key exchange

We will use Andrea's calculator for EC multiplication over a finite field and follow the procedure in Cryptography Homework 7 example.docx.

Pick public information

Select the curve $y^2 = x^3 + 5x + 3$ with a modulus, p, of 97. Together, select a point that will be the base point, P (P has two numbers, the x and y coordinates.) The curve, modulus (Field in Andrea's calculator), and base point will be public knowledge. Record them.

Curve: a = 5, b = 3 ($y^2 = x^3 + 5x + 3$)   Modulus (or Field): 97
P: x __________ y __________



Curve: a 5 b 3

Field: p 97

n: n

P: x y

$Q = n \cdot P$: x y

You and partner pick x and y for P
It becomes public information

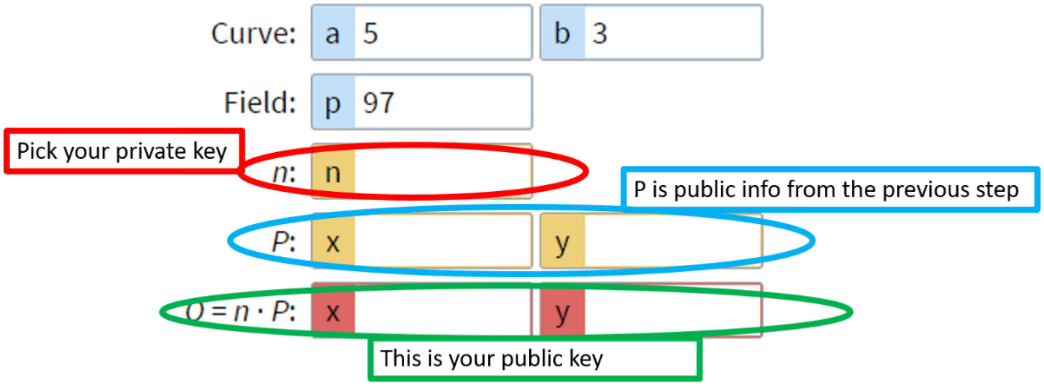
The Curve, Field (or modulus), and the x and y coordinates of P are public information; everyone knows them.

Pick Public Keys

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, field (modulus), P, and the coordinates Q that you computed.)

Alice
Select n ____________ Alice's private key
Compute Q ____________ Q = nP Alice's public key
Alice gives public key to Bob, keeps private key secret

Bob
Select n ____________ Bob's private key
Compute Q ____________ Q = nP Bob's public key
Bob gives public key to Alice, keeps private key secret



Curve: a 5 b 3

Field: p 97

n: n

P: x y

$Q = n \cdot P$: x y

Pick your private key

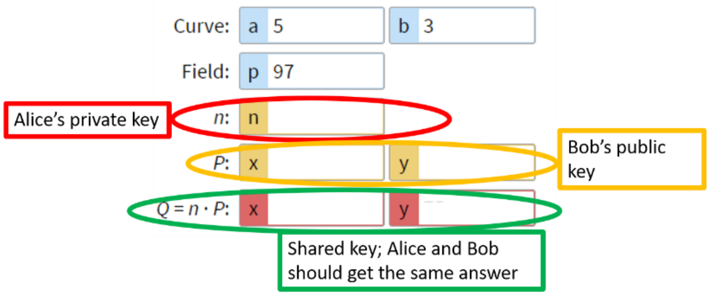
P is public info from the previous step

This is your public key

Compute the shared key

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.

Alice



Curve: a 5 b 3

Field: p 97

n: n

P: x y

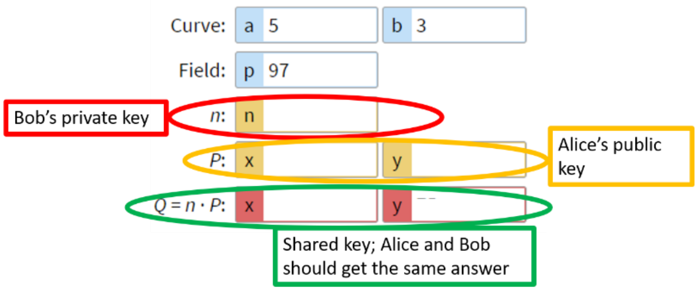
$Q = n \cdot P$: x y

Alice's private key

Bob's public key

Shared key; Alice and Bob should get the same answer

Bob



Curve: a 5 b 3

Field: p 97

n: n

P: x y

$Q = n \cdot P$: x y

Bob's private key

Alice's public key

Shared key; Alice and Bob should get the same answer

Alice's shared key ____________

Bob's shared key ____________

Alice and Bob should both have the same shared key.

Security

The attacker Eve must be able to compute either Alice's or Bob's private key to break the encryption. If the numbers are huge, this is exceedingly difficult.

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." You will use a different curve in the next part, but you will need to be able to find that same line to find the number of points on the curve and the number of points in the subgroup.

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 points. If the base point you select is in a small subgroup, your security is weakened.

Set your calculator to the curve $y^2 = x^3 + x + 2$, 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)
Look at the line, “The subgroup generated by P has __ points.” Write that number down for each of the points above.

For one or two of the points, set n to 1. Then increment n with the up arrow and watch how the point Q jumps around on the graph to the left. What happens when the subgroup has a small number of points?

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 (optional)

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 do not 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 $y^2 = x^3 + x + 2$, with modulus $p = 97$? What are the factors of that number? Compare the factors to the subgroup sizes you found for the points.

3) Which point on the curve in question 2 provides the best security when it is chosen as a base point (P)? I.e., the subgroup has the largest number of points? Which one provides the worst security.

4) From the Choose your own paragraph, submit the curve, modulus, base point, and number of points in the subgroup. (Optional)