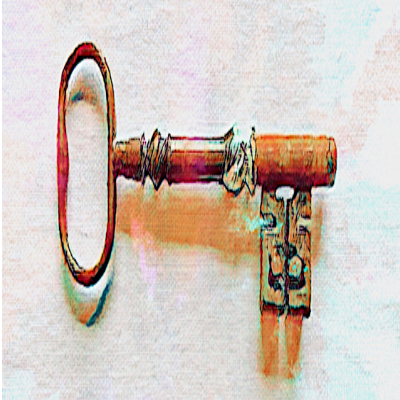
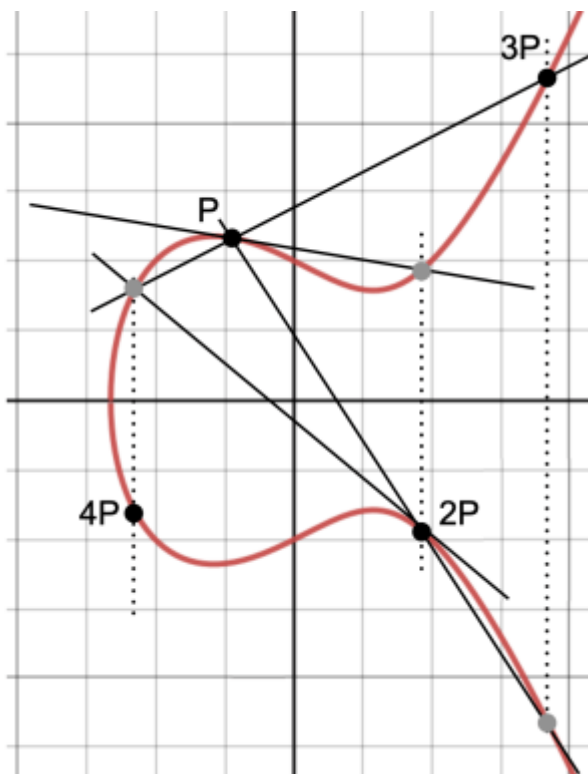




Key exchange: It's a wrap!

Description

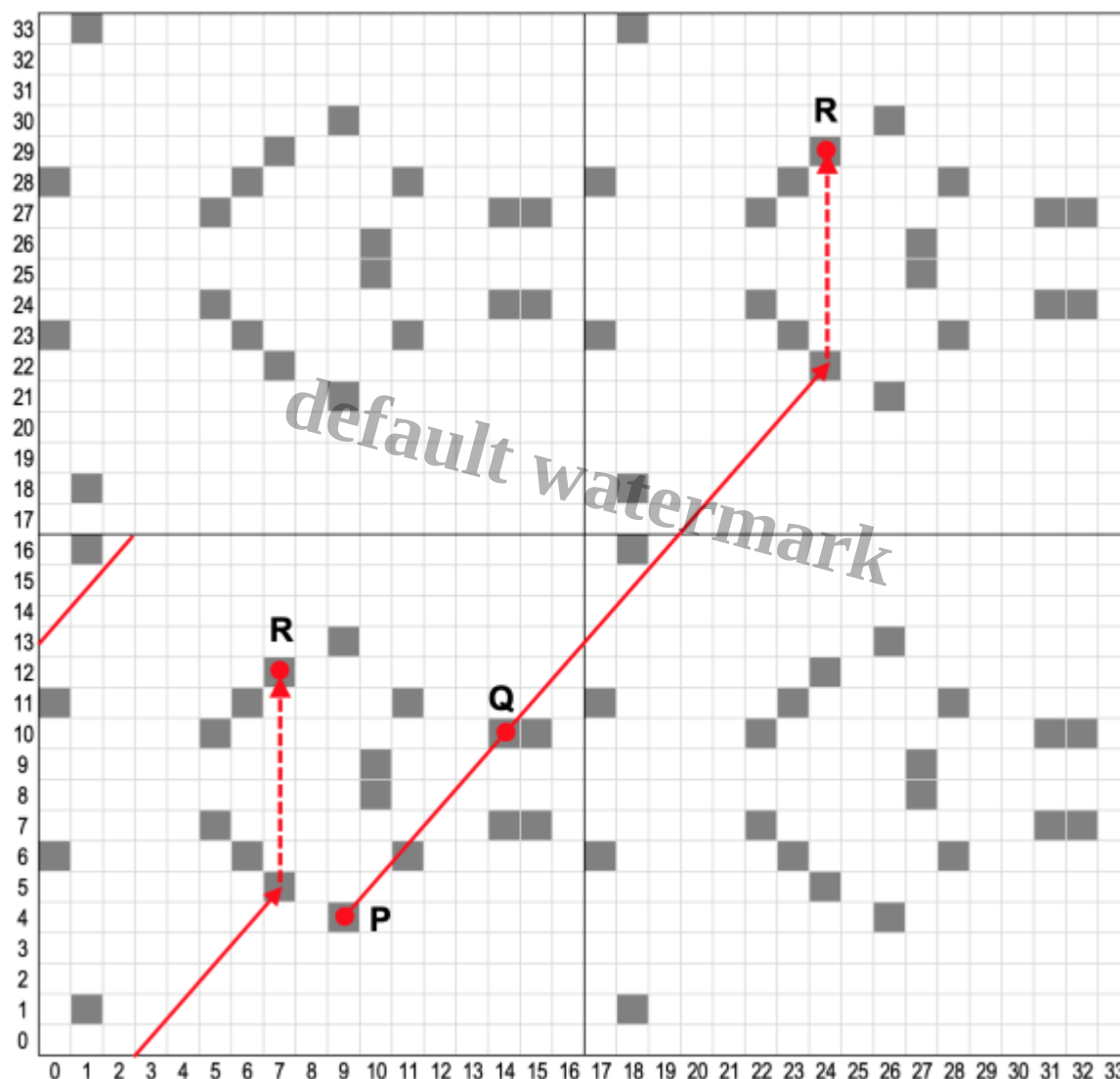
In an earlier post ([Elliptic trapdoors](#)) I did my best to explain addition and multiplication along an elliptic curve. The diagram showed a point P doubled to produce $2P$, then added to itself to produce $3P$, and added again to produce $4P$. The dotted lines indicated the mirror reflection operation to arrive at the final value. The construction looked like this.



I described the finite field translation of the graph in [my previous post](#). Here's the chart of

$$y^2 \bmod 17 = (x^3 - 2x + 2) \bmod 17$$

showing values of x and y across four 17×17 frames. The chart extends indefinitely in all directions. Here I show in red the calculation for $P + Q = R$. The steps can all take place in just one frame (bottom left). As the red line leaves a frame it re-enters at the same point on the opposite side of the frame. That wrapping operation could repeat many times, depending on the size of the space traversed by the red line.



What is the red line? It's a vector connecting two points added and projected in either direction until it meets (precisely) another cell in the field. The maths is straight forward geometry: calculate the slope of the vector, then test it against each point in the field to see if it is on the vector. Then find the point reflected across the mid axis of the frame. The process parallels the operations applied to the elliptic curve for addition.

Diffie-Hellman Elliptic-Curve (DHEC) Key Exchange

Mark Hughes provides a very helpful explanation of the [Diffie-Hellman Elliptic-Curve Key Exchange](#) method with diagrams. I explored in a previous post ([Sharing a secret number](#)) a process by which two people might exchange a secret code number across the public Internet, assuming someone else might

be listening in. Each of the two people communicating retains a private key that they don't disclose to anyone. There's also a public key that anyone can read, and will be instrumental in encrypting a message, but not decrypting it, unless they have the private key as well.

If you and I are to communicate in secret then we need to agree to use the same Elliptic curve, with the same parameters. We also pick a random point on the curve, i.e. a point in the finite field chart above. It will with certainty be an integer, and correspond with a point on the elliptic curve via the translation described in the previous post (Elliptic fields). (I'll use Hughes' nomenclature here.)

I choose a secret integer $\hat{1}$. You choose $\hat{2}$. I multiply the point G by itself $\hat{1}$ times, and you multiply the point G by itself $\hat{2}$ times.

We each arrive at new points $A=\hat{1}G$, and $B=\hat{2}G$ that we exchange with one another.

Starting at the new points, we each multiply our new points by our own secret numbers using the field method described above to yield the secret message. Lest we think this process is easy to implement and/or codebreak by hand, Hughes reminds us:

Remember that we used small numbers to make the process easier to understand. DHEC uses a publicly known equation with large coefficients and modulus, for example, curve1559, which might very well be securing your browser right now.

max:

115792089210356248762697446949407573530086143415290314195533631308867097853951

curve: $y^2 = x^3 + ax + b$

$a=115792089210356248762697446949407573530086143415290314195533631308867097853948$

$b=$

41058363725152142129326129780047268409114441015993725554835256314039467401291

Summary

A great deal of modern cryptography is based upon the Diffie-Hellman exchange, which requires that two parties combine their messages with a shared secret that is difficult for a bad actor to deduce.

Elliptic-curve Diffie-Hellman allows microprocessors to securely determine a shared secret key while making it very difficult for a bad actor to determine that same shared key.

The multipliers our computers choose for our private keys $\hat{1}$ and $\hat{2}$ will also be many digits long as will the coordinates of the initial point G, further securing the method.

Bibliography

- Hughes, Mark. 2019. How Elliptic Curve Cryptography Works. *All About Circuits*, 26 June. Available online: <https://www.allaboutcircuits.com/technical-articles/elliptic-curve-cryptography-in-embedded-systems/> (accessed 15 May 2021).

Note

- The diagram of the grid above is my own interpretation of the process.

Category

1. Architecture

Tags

1. cryptography
2. Diffie-Hellman
3. Elliptic curve

Date Created

June 19, 2021

Author

rcoyne99

default watermark