ELLIPтиЧНьE CURVE CRYPTOGRAPHY

Public-key cryptography based on elliptic curves is gradually replacing RSA thanks to faster implementations and smaller key sizes.

“CLOCK CRYPTOGRAPHY”
Diffie-Hellman using points on a circle
Imagine Alice and Bob want to arrive at a shared secret point on a “clock”. They pick starting point, the “generator” (called “g”), and each pick their own secret exponent (we’ll call these a and b). We also pick a modulus: 12 hours for a real clock, but a prime for a crypto-clock.

<table>
<thead>
<tr>
<th>Base Point</th>
<th>Public Key</th>
<th>Shared Secret</th>
</tr>
</thead>
<tbody>
<tr>
<td>g=3 mod 17</td>
<td>A=10 mod 17</td>
<td>b=5 mod p</td>
</tr>
</tbody>
</table>

Alice picks a private key a=10 and raises the generator g=3 to the power of her private key modulo p=17. She arrives at a point on the curve, then bounces towards the x-axis. She stops at the point C. Bob picks a private key b=5, and wants to derive a shared secret with Alice. He raises her public key A=10 to the power of his private key 5 modulo p=17. The resulting point (5,9) is a shared secret.

Bob can multiply Alice’s public point by his private scalar to reach a secret point shared with Alice.

ELLIPтиЧНьE CURVES
Diffie-Hellman using points on a surreal “clock”
Like clock cryptography, elliptic curve cryptography relies on the ideas of a base point (the “generator” in clock cryptography) and a prime modulus, but the circle is replaced with an algebraic curve which is scattered over something known as a prime field (i.e. a finite field).

POINT ARITHMETIC
To replicate the same ideas as clock cryptography using elliptic curves, we’ll need a way to add points on an elliptic curve just like we’d add points on a circular “clock”. Once we can add points together, we can build a “scalar multiplication” function which lets us combine a base point and secret key (a big number, a.k.a. “scalar”) to get a point on the curve which represents a public key. But before we can multiply, we first need to be able to add.

POINT ADDITION OVER REAL NUMBERS
This diagram represents adding the points A and B on an elliptic curve. It works kind of like a game of billiards where the ball always bounces towards the x-axis.

We first draw a line from A to B, and where that line intersects with the curve, we “bounce” towards the x-axis. We intersect with the curve again.

The resulting point, C, is considered the sum of A and B on the curve. Think of it like adding together hours on the crypto clock above, and how their combination is a point on the elliptic curve.

POINT ADDITION OVER A FINITE FIELD
The elliptic curves used in cryptography are scattered over a prime field, and do not look like the curve above, but rather a speckling of points. However, underlying these dots is something with the same properties as a curve like above. We are able to add points in a similar manner, by drawing a line (with the wrapping behavior you see) from A to B, continuing until we intersect with another point on the curve, then “bouncing” vertically as we did before until we intersect with the curve again at point C.

SCALAR MULTIPLICATION
Once we’re able to perform point addition, we can construct a scalar multiplication operation. This involves adding a base point to itself repeatedly, where the number of times we do this is “scalar” input to the multiplication operation. In practice, this scalar represents an elliptic curve private key.

We now have something that works a lot like clock cryptography above: we can pick a curve and a standard base point on that curve. Alice can pick a private scalar value for her secret key, and multiply her scalar by the base point to find a point on the curve that represents her public key. Bob can multiply Alice’s public point by his private scalar to reach a secret point shared with Alice.