by @bascule

ELLIPTIC CURVE CRYPTOGRAPHY

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

"CLOCK CRYPTOGRAPHY" VS ELLIPTIC CURVES

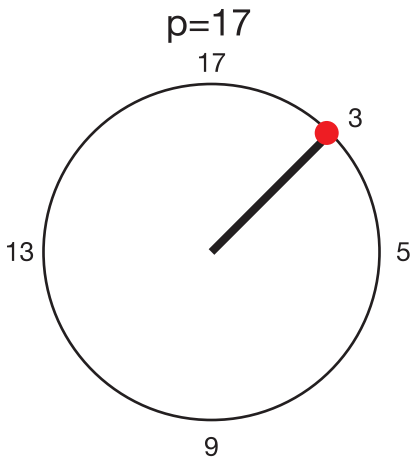
"CLOCK CRYPTOGRAPHY"

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.

Base Point

g=3
p=17

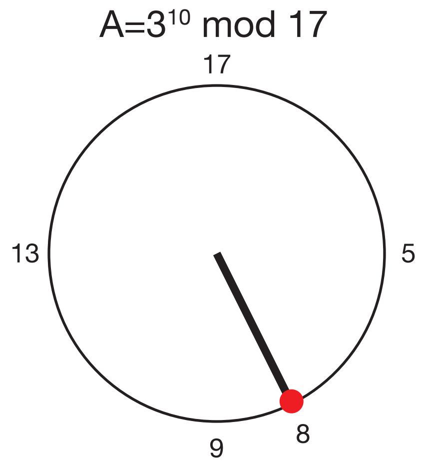


We need a base point (a.k.a generator) whose exponents are uniformly distribut around the clock.

- g=1 is bad because $1^n=1$
- g=2 is not a primitive root modulo 17
- g=3 is a primitive root modulo 17

Public Key

a=10
$A=3^{10}$ mod 17

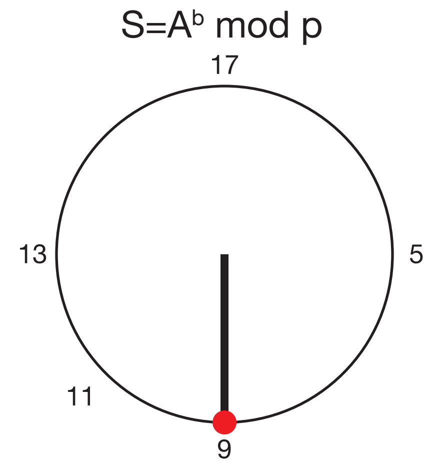


Alice picks a private key a=10 and raises the generator g=3 to the power of her private key modulo p=17.

The value she arrives on, A=8, acts as her public key.

Shared Secret

b=5
$S=A^b$ mod p



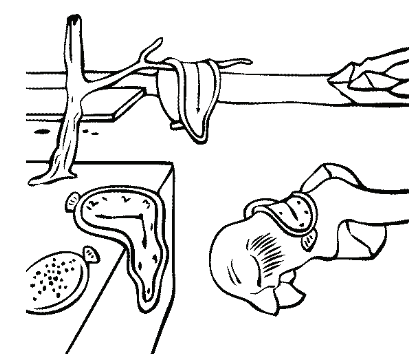
Bob picks a private key b=5, and wants to derive a shared secret with Alice. He raises her public key A=8 to the power of his private key 5 modulo p=17. The resulting point ("9") is a shared secret.

ELLIPTIC CURVES

Diffie-Hellman using points on a surreal "clock"

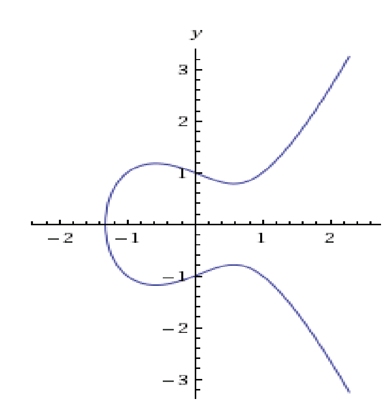
Like clock cryptography, elliptic curve cryptography relies on the ideas of a base point (the "generator" in clock cryptography) and a prime modulus, but the circle is replaced with an algebraic curve which is scattered over something known as a prime field (i.e. a finite field)

Distorted Clocks



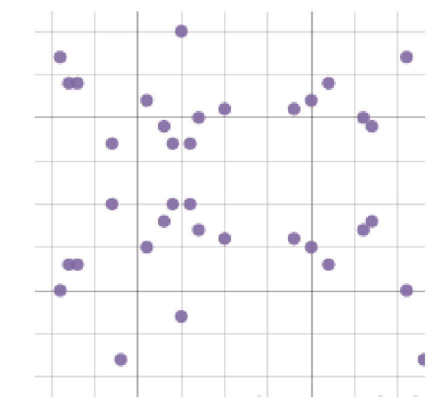
Elliptic curves work a bit like a clock in a Salvador Dali Painting. Like in the painting, the clock is bent and distorted. This prevents index calculus attacks which work against clock cryptography and therefore allows for use of smaller keys.

Not An Ellipse



They sound like they should look like ellipses, but instead they're symmetrical curves with legs stretching off to a point at infinity. However, we can still add points on them much in the same way we can add points on a clock.

Finite Fields

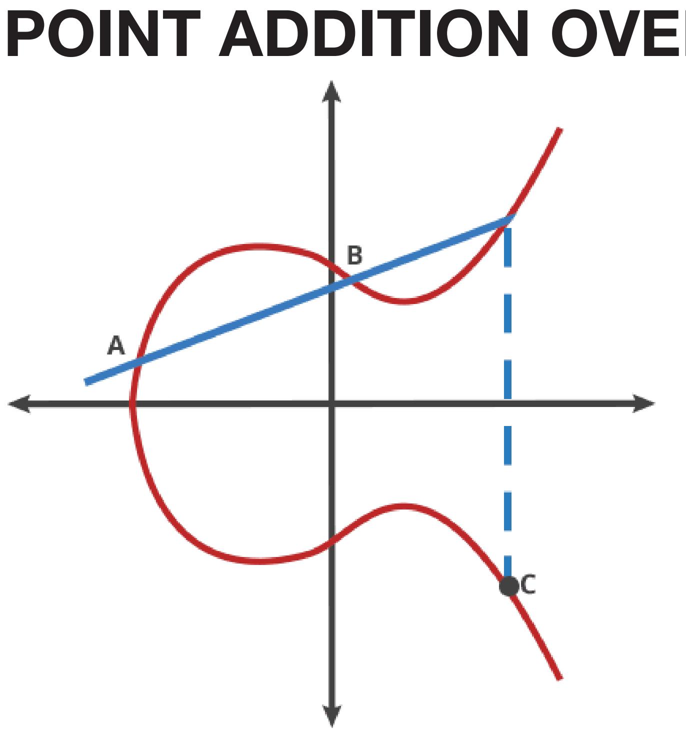


When plotted over a finite field, elliptic curves appear to look a bit like a random scatter plot. However, we can still reason about them as curves. They also retain the same symmetry as their "real" counterparts.

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

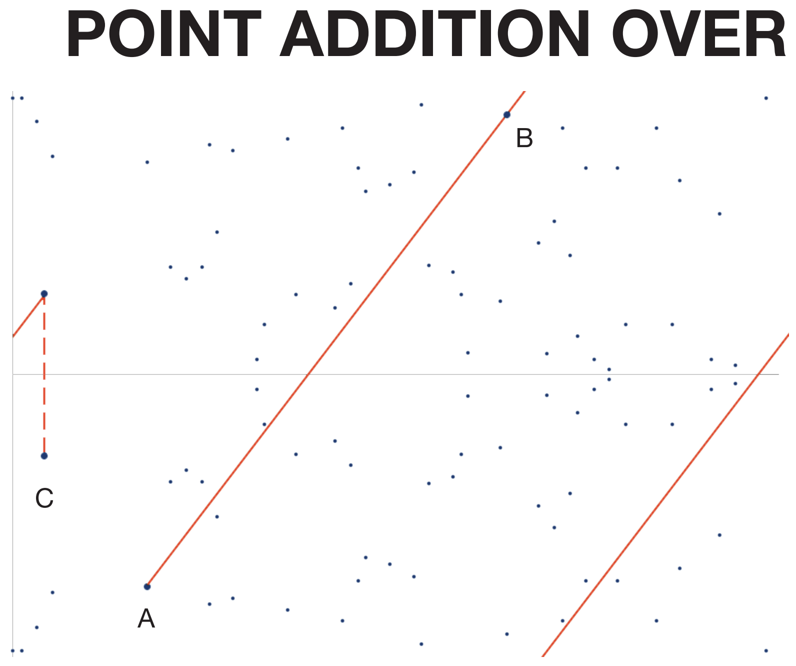


This diagram represents adding the points A and B on an elliptic curve. It works kind of like a game of billiards where the ball always bounces towards the x-axis.

We first draw a line from A to B, and where that line intersects with the curve, we "bounce" towards the x-axis until 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

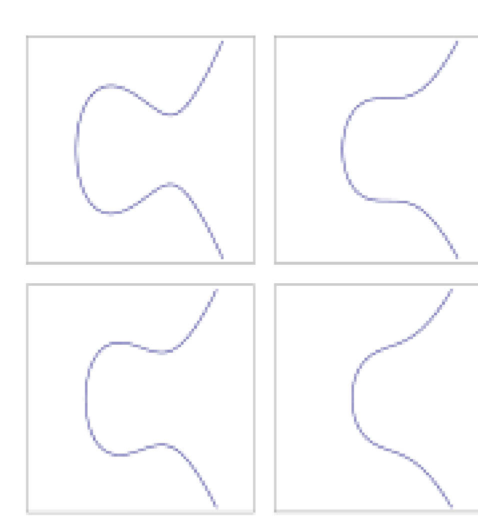
Once we're able to perform point addition, we can construct a *scalar multiplication* operation. This involves adding a base point to itself repeatedly, where the number of times we do this is "scalar" input to the multiplication operation. In practice, this scalar represents an elliptic curve private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

CURVE FORMS

There are several different forms of elliptic curves used in cryptography, each corresponding to the name of the mathematician who discovered it. Different curve forms are used for different applications, however all curve forms can be converted to the other forms.

WEIERSTRASS: NIST CURVES AND BRAINPOOL

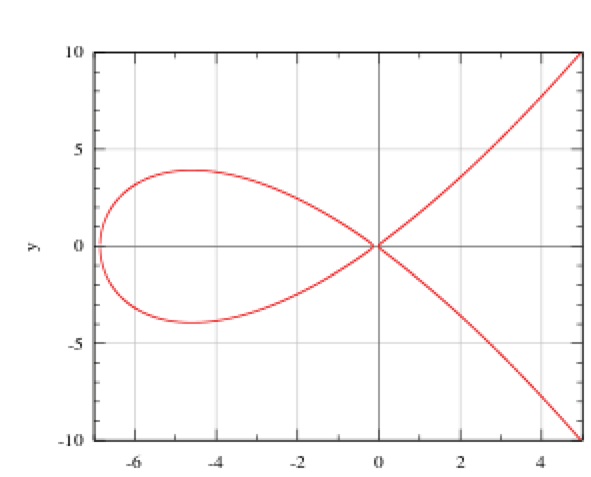


Weierstrass curves are described by the equation $y^2 = x^3 + ax + b$ (specifically this is the "short" form of the Weierstrass equation).

These curves were the most popular until recently, standardized by NIST (P-192, P-224, P-256, P-384, P-521) and Brainpool.

However, due to the complexity of the associated field arithmetic and its error-prone nature, it has generally lost favor.

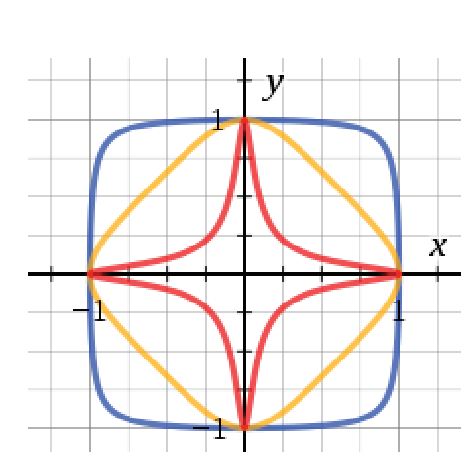
MONTGOMERY: Curve25519



Montgomery curves are a newer form described by the equation $By^2 = x^3 + Ax^2 + x$. The most popular is Curve25519, used by the "X25519" Diffie-Hellman function.

Montgomery curves are attractive because of the "ladder" method of scalar multiplication, a simple, fast approach which is easy to implement correctly (i.e. in constant-time). The Montgomery ladder only takes a single coordinate as input, eliminating a whole class of attacks present in Weierstrass when points aren't on the curve.

EDWARDS: Ed25519 AND Ed448-GOLDILOCKS



Edwards curves are the newest form of elliptic curve, and are described by the equation $x^2 + y^2 = 1 + dx^2y^2$.

Edwards curves are one of the main focus areas of ECC research and standardization. They are particularly interesting when used with variants of the Schnorr digital signature algorithm.

Ed25519, the Edwards form of Curve25519 for use with the EdDSA digital signature algorithm, is the most popular Edwards curve today. Ed448-Goldilocks is another Edwards curve that has been receiving recent attention.