Elliptic Curve Cryptography

Eli T. Drumm, 2014 October 20

Elliptic Curves

An elliptic curve is the set of solutions (x, y) of an equation of the form

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6.$$

x, y, and the a_i are values in some field $\mathbb{F}$, which can be $\mathbb{R}$ or $\mathbb{C}$, for example. For cryptography, we are interested in the case where $\mathbb{F}$ is a finite field; in particular, either $\mathbb{F}_p$ where p is a prime or $\mathbb{F}_{2^m}$ for some m, "binary fields". For an illustration of the basic idea we focus on the first case.

If the characteristic of the underlying field is not 2 or 3, the equation can be simplified to $y^2 = x^3 + ax + b$. For a given elliptic curve E, we can take the points on the curve in our chosen finite field $\mathbb{F}_p$ together with a special *point at infinity* $\mathcal{O}$ which can be thought of as lying simultaneously on top of and below all vertical lines in the plane. Formally, the underlying set $E(\mathbb{F}_p)$ of points in our group is defined as

$$E(\mathbb{F}_p) = \{(x, y) \in \mathbb{F}_p^2 | y^2 = x^3 + ax + b\} \cup \{\mathcal{O}\}.$$

To use elliptic curves in cryptographic algorithms, we are going to need Abelian groups from curves. The group addition algorithm should be, of course, a way of producing a third point from any two points on the curve. There is an explicit addition algorithm for computing with coordinates (consult the resources) but the intuitive explanation is geometric and is best seen using curves over the reals (this way you deal with pretty curvy pictures rather than isolated dots).

Given two points on a curve, connect them with a line. Find the third point on the curve that intersects the line (there will be one), and then take the symmetric point on the opposite side of the x-axis. *This* point is the result of the addition. (In the case that a point is intersected by a line tangent to the curve at that point, it counts as two points.) The addition of any three collinear points on the curve results in $\mathcal{O}$. $\mathcal{O}$ acts as the identity of the group.

To see this actually forms an Abelian group, we have to check that the addition has the requisite properties. We have an identity, $\mathcal{O}$. Any line touching the curve in two points does in fact have a third point of intersection, so that addition is well-defined (this follows from a theorem about elliptic curves). Every point has an inverse, the symmetric point opposite the x-axis. Commutativity holds because any collinear points are collinear regardless of which order you consider them. The tricky bit is the last one, associativity, verifying that for any points P, Q, R on E, $(P + Q) + R = P + (Q + R)$. (The straightforward proof of this requires some tedious computation, but it does work out.)

Curves in Cryptography

As with the multiplicative groups we have seen in the course, we can define the discrete logarithm problem (computational or decisional) for elliptic curves. Given a curve E, a base point on the curve P, and a point aP, which is P added to itself a times using the group addition, find a. The discrete logarithm problem for elliptic curves, which is believed to be intractible in the general case, underlies the application of elliptic curves in cryptography. Many of the algorithms and protocols using elliptic curve groups are modifications of their "traditional" versions such as Diffie-Hellman, ElGamal, the Digital Signature Algorithm, etc.

To carry out a procedure like Diffie-Hellman key exchange on an elliptic curve, Alice and Bob need to be working with the same curve. To select curves, one option is to use a specific curve that has been selected and published in some standard backed by an organization. Example: secp256k1, the curve used in the ECDSA implementation in Bitcoin. If the domain parameters for the curves are chosen carefully and safely, this potentially allows the use of curves with properties that are especially fast/efficient to compute with.

At the same time, even if a curve looks random, and your favorite government standards organization says it's *definitely* random, maybe you want to take that assurance with a grain of salt (and only use standard curves with parameters selected in a verifiably random or justifiably reasonable way that decreases the likelihood that someone somewhere has a back door).

An alternative is to use some specified procedure to produce random curves that will only be susceptible to known pathological attacks with negligible probablility.

Advantages

Given that ellptic-curve algorithms don't allow us to perform any fundamentally new cryptographic feats, it's reasonable to ask why we might want to use elliptic curves in cryptography in the first place. The primary reason is (under commonly held assumptions) when compared with RSA, equivalent security requires shorter keys and the implementing algorithms run faster. The differences aren't always a full order of magnitude for the most part, but are certainly enough to warrant using "EC" versions of cryptographic standards when computational resources are more precious (for instance on mobile devices).

One other reason is that while the discrete logarithm problem for elliptic curves has remained firmly intractible, computing discrete logs for the multiplicative groups used in standard Diffie-Hellman and the like has seen more progress recently. From this perspective, elliptic curves are insurance against this progress continuing in the future.

Resources

For further reading, there are some lighter overviews as well as some comprehensive textbooks that cover all the mathematical background and technical details of implementation and then some. As far as books go: L. Washington's *Elliptic Curves: Number Theory and Cryptography, 2nd ed.* (2008) and Hankerson, Menezes Vanstone's *Guide to Elliptic Curve Cryptography* (2004) are two I used to prepare this. Also, there are some thrilling standards that are just absolute must-reads, like "The Elliptic Curve Digital Signature Algorithm (ECDSA)" by Johnson, Menezes Vanstone, "SEC 2: Recommended Elliptic Curve Domain Parameters" published by Certicom Research, and of course "FIPS PUB 186-4, FEDERAL INFORMATION PROCESSING STANDARDS PUBLICATION, Digital Signature Standard (DSS)" published by the Information Technology Laboratory of the National Institute of Standards and Technology, part of the US Department of Commerce... riveting stuff. Also, Wikipedia.