Elliptic Curve Cryptography

Elaine Brow, December 2010
Math 189A: Algebraic Geometry

1. Introduction to Public Key Cryptography

To understand the motivation for elliptic curve cryptography, we must first understand the purpose of *public key cryptography* as a whole. To do this, we introduce a hypothetical situation involving two old friends of cryptographers everywhere, Alice and Bob.

Example 1.1. (Alice and Bob) Suppose Alice and Bob would like to communicate secret messages to each other. The only problem is, everyone knows that an evil eavesdropper (appropriately named Eve) has access to all communication between Alice and Bob. How can they tell their secrets without Eve hearing (or at least without Eve hearing anything of significance), and keep her from tampering with the information on its way from one person to another?

This is where the idea of *public keys* comes in. Alice and Bob each have a key, some number or mathematical procedure that can be applied to messages, composed of a public piece and a private piece. The private pieces of these keys are never transmitted, while the public pieces are accessible to everyone, including Eve. When Alice wants to send a message to Bob, she uses the public piece of Bob's key to encrypt the information, and sends it without worrying at all about who sees it. Then Bob uses the private part of his key to decrypt the information. Because Bob is the only one who has the private part of his key, he is the only one who can decrypt it. For additional security, Alice and Bob may also have public and private signatures, which work similarly to the keys. If Alice wants Bob to know that the message he receives from her is authentic, she'll apply a private signature to some authentication message before sending it; when Bob wants to know that it's hers, he'll apply the easily accessible public part of her signature to that, which will return the authentication. If Eve tampers with the signature, it will return garbled, and Bob will know it is corrupted.

If the mathematical public and private keys don't make sense, just think of the public keys as padlocks and the private keys as the keys to those locks. Alice and Bob both publicly distribute copies and copies of each of their locks, but always keep the key safely with them. Then to send Bob a message, Alice just has to find one of Bob's padlocks, lock her message up with it, and send it to him. He has the single key to all of his locks, so he is only one who can open it.

Notice that the security of this system does not rely at all on Alice and Bob finding a secure way to transmit information, but it relies very heavily on Alice and Bob each having private keys that are very, very difficult to retrieve using only their public keys. Eve can only be thwarted if the information that she can intercept is totally useless. This brings us to the *elliptic curve discrete logarithm problem*, which we will see can be made sufficiently difficult to give us a useful pair of keys. First we must explain elliptic curves.

2. The Elliptic Curve Group Law

Definition 2.1. An *elliptic curve* is a nonsingular projective algebraic curve over some field k with genus 1 and a specified point $\mathcal{O}$ (this will be the "point at infinity"). So long as k does not have characteristic 2 or 3, this will be a smooth plane cubic curve with the point at infinity, and we can describe the curve as points satisfying the equation

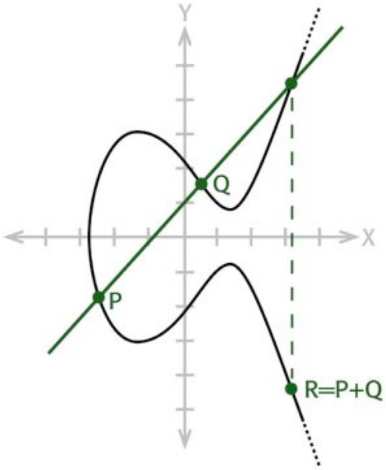
$$y^2 = x^3 + ax + b,$$

with a and b such that the discriminant,

$$\Delta = -16(4a^3 + 27b^2),$$

is nonzero (which will give the desired nonsingularity).

The group law on an elliptic curve. The operation exploited for key selection in elliptic curve cryptography comes from considering the elliptic curve as an abelian group with points as elements. The group law is point addition; to add two points P and Q, we will draw the line PQ through them (or use the tangent line at P to add it to itself), find the third point of intersection $-R$ of that line, and reflect it over the axis of symmetry of the curve. The resulting point, R, will be the sum of P and Q. For the purposes of this addition, note that the point at infinity $\mathcal{O}$ lies on any line through a point and it's opposite. The formal properties of the addition law are described below.



Theorem 2.2 The addition law on elliptic curve C has the following properties (where $\mathcal{O} = -\mathcal{O}$ is the point at infinity, and if $P = (x_0, y_0)$, then $-P = (x_0, -y_0)$):

(i) For point $P \in C$, $P + \mathcal{O} = P$,

(ii) For points $P, Q \in C$, $P + Q = Q + P$

(iii) For point $P \in C$, there is some point $-P$ such that $P + (-P) = \mathcal{O}$

(iv) For $P, Q, R \in C$, $(P + Q) + R = P + (Q + R)$.

In short, the addition law gives us the group properties that we desire. Additionally, we will note that the subset of points in this group whose both coordinates belong to a given field k, along with the point at infinity, will form a subgroup of the curve group C. This will be important, because the curves used in elliptic curve cryptography are defined over a finite field, and we need that set to be closed under point addition.

Because our goal now is not to construct elliptic curve cryptography, but rather to understand how it works, we will omit the formal proof, but notice that most of the properties above follow directly from the geometric description of point addition.

3. The Elliptic Curve Discrete Logarithm Problem

Now that we understand the properties of elliptic curves as groups, we can approach the elliptic curve discrete logarithm problem, from which elliptic curve cryptosystems draw their strength.

Definition 3.1 The elliptic curve discrete logarithm problem (ECDLP) is this: *given an elliptic curve C defined over $\mathbb{F}_q$ and two points $P,\ Q \in C$, find an integer x such that $Q = xP$.*

It can be understood on a very elementary level why this problem might be difficult to solve. Imagine going through several iterations of the point adding process described above on a curve that has many, many points, then erasing all of the intermediate steps. It is not immediately apparent how to proceed when trying to recreate the process you have just made invisible.

In fact, nobody knows exactly how difficult this problem is to solve, because no one has come up with an efficient algorithm to solve it. It is, however, believed to be more difficult to solve than the general discrete logarithm problem, and the various factorization problems that are used in other cryptosystems (and the best methods for cracking these problems do not seem to adapt easily to elliptic curve problems), which suggests that elliptic curve cryptography is the strongest of all the available cryptographic systems. Looking at the required key sizes for multiple given levels of security (where “more secure” means “takes longer to break”) of elliptic curve cryptosystems as compared to other traditional cryptosystems, the required key sizes of other systems rise exponentially as difficulty increases, while the increase in required key size for elliptic curve systems is relatively miniscule.

What this means is that if we set up our cryptosystems so that they can be cracked only by solving ECDLP, Bob and Alice’s messages will be extremely secure.

4. Examples of Elliptic Curve Cryptosystems

Because the ECDLP tells us that for $Q = xP$, x is very difficult to find, we want Q and P to be the public key in any cryptosystem we use (the padlocks) and x to be the private key (the hard-to-manufacture key to the padlocks). There are multiple ways to construct cryptosystems that operate this way, so we will provide two as examples. Both are elliptic curve analogues of preexisting cryptosystems that were created to use the general discrete logarithm problem; adaptation is easy since the structure of the ECDLP is so similar to that of the original DLP. Both also assume some existing system of embedding messages into points on the elliptic curve. There are a number of ways to do this, none of which are specifically attached to the given cryptosystems, so we just assume that we have chosen some embedding of message m into point P_m, and that this embedding is publicly known (so that Bob can retrieve the embedded message once he obtains P_m). Our first example is an adaptation of the ElGamal public key cryptosystem:

Example 4.1 (The ElGamal Elliptic Curve Cryptosystem) Suppose that we have some elliptic curve C defined over a finite field $\mathbb{F}_q$ where $q = p^n$ is large (and p is prime). Suppose that C, q, and a point $G \in C$ are publicly known, as is the embedding system $m \mapsto P_m$. When Alice wants to communicate secretly with Bob, they proceed thus:

- Bob chooses a random integer b, and publishes the point bG (while b remains secret).
- Alice chooses her own random integer a and sends the pair of points $(aG, P_m + a(bG))$ to Bob (while a remains secret).
- To decrypt the message, Bob calculates $b(aG)$ from the first part of the pair, then subtracts it from the second part to obtain $P_m + a(bG) - b(aG) = P_m + abG - abG = P_m$, and then reverses the embedding to get back the message m.
- Eve, who can only see bG, aG, and $P_m + a(bG)$ must find a from aG or b from bG to make sense of $P_m + a(bG)$, so her problem is reduced to the ECDLP, and she is thwarted.

This is a successful cryptosystem because every operation that Alice and Bob have to carry out (addition and subtraction on the curve) is relatively easy, while the operation that Eve would have to perform to crack the system is extremely difficult (or for real-life villains without the proper resources, perhaps impossible). Our next example, an analogue of the Massey-Omura public key cryptosystem, operates on a similar back-and-forth series of easy problems for Alice and Bob that produces the ECDLP for Eve.

Example 4.2 (The Massey-Omura Elliptic Curve Cryptosystem) As before, suppose that we have some elliptic curve C defined over a finite field $\mathbb{F}_q$ where $q = p^n$ is large, and $N = |C|$. (We did not need the number of points on C in the previous example, but this information is never relied on to be secret, because it can be calculated with relative efficiency.) The embedding system $m \mapsto P_m$, as well as q, C, and N, are publicly known. When Alice wants to communicate secretly with Bob, they proceed thus:

- Alice chooses a random integer c such that $0 < c < N$ and $gcd(c, N) = 1$, and sends cP_m to Bob (while keeping c secret).
- Bob then chooses a random integer d with the same properties as c, and sends $d(cP_m)$ back to Alice (while keeping d secret).

- Alice can find c' from c such that $cc' \equiv 1$, and sends $c'(d(cP_m)) = dP_m$ back to Bob.
- To decrypt the message, Bob multiplies $d'(dP_m)$, where $dd' \equiv 1$, and retrieves P_m, and reverses the embedding to get the message m.
- Again, Eve's problem reduces to the ECDLP because she would need to retrieve c, d, or dc from cP_m, dcP_m, or dP_m, etc., to steal the message at any point.

Still, Alice and Bob's calculations are much, much easier to perform than Eve's ECDLP, so this cryptosystem succeeds.

5. Conclusion

The examples that we have provided above are conceptually simple, but it is important to remember that the groups we are considering are enormous, the keys are large enough to be measured in number of bits, and the operations that are described as "relatively easy" are called that because very fast computers can complete them in fairly reasonable amounts of time (and the "difficult" problems are then so difficult that those same computers cannot solve them in any reasonable amount of time).

We must also remember that not every choice of curve, finite field, and point is created equal. In the ElGamal system, Alice and Bob are operating not on the entire curve, but on the cyclic group generated by G, and in the Massey-Omura system, that generated by P_m. These decisions must be made well, and it would take much more space than this to explain how to make them well.

However, with the appropriate choices, we do get to see the power of something that is quite simple, in the elliptic curve group law. A geometric idea as simple as connecting dots (though one dot does have to lie at infinity) gives us a secret messaging system that has yet to be cracked.

References

[1] Darrel Hankerson. *Guide to Elliptic Curve Cryptography.* Springer, 2004.

[2] Neal Koblitz. Elliptic curve cryptosystems. *Mathematics of Computation*, 48:203 – 209, 1987.

[3] Martin Leslie. Elliptic curve cryptography. (An ECC research project), 2006.

[4] Miles Reid. *Undergraduate Algebraic Geometry.* Cambridge University Press, 1988.

[5] Matthew Simpson. http://math.rice.edu/ hargis/vigre/. (Image source).