Quantum algorithms (CO 781, Winter 2008)
Prof. Andrew Childs, University of Waterloo

LECTURE 3: Quantum attacks on elliptic curve cryptography

In the last lecture we discussed Shor's algorithm, which can calculate discrete logarithms over any cyclic group. In particular, this algorithm can be used to break the Diffie-Hellman key exchange protocol, which assumes that the discrete log problem in $\mathbb{Z}_p^\times$ (p prime) is hard. However, Shor's algorithm also breaks *elliptic curve cryptography*, the main competitor to RSA. In this lecture we will introduce elliptic curves and show how they give rise to abelian groups that can be used to define cryptosystems.

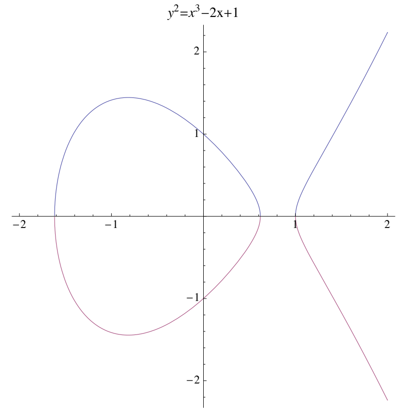
This lecture is only intended to be a survey of the main ideas behind elliptic curve cryptography. While breaking such cryptosystems is a major potential application of quantum computers, only a few implementation details differ between the algorithms for discrete log over the integers and over elliptic curves; no new quantum ideas are required.

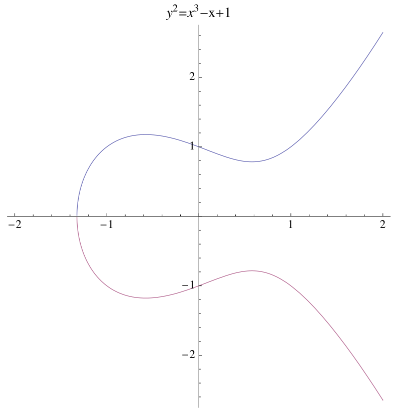
Elliptic curves Fix a field $\mathbb{F}$ whose characteristic is not equal to 2 or 3. (Cryptographic applications often use the field $\mathbb{F}_{2^n}$ of characteristic 2, but the definition of an elliptic curve is slightly more complicated in this case, so we will not consider it here.) Consider the equation

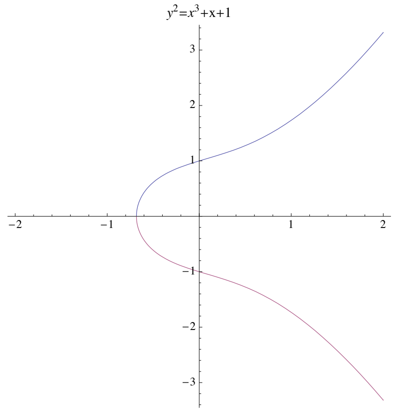
$$y^2 = x^3 + ax + b \tag{1}$$

where $a, b \in \mathbb{F}$ are parameters. The set of points $(x, y) \in \mathbb{F}^2$ satisfying this equation, together with a special point $\mathcal{O}$ called the *point at infinity*, is called the *elliptic curve* $E_{a,b}$. A curve is called *nonsingular* if its *discriminant*, $\Delta := -16(4a^3 + 27b^2)$, is nonzero, and we will assume that this is the case for all curves we consider.

Here are a few examples of elliptic curves over $\mathbb{R}^2$:







Such pictures are helpful for developing intuition. However, for cryptographic applications it is useful to have a curve whose points can be represented exactly with a finite number of bits, so we use curves over finite fields. For simplicity, we will only consider the case $\mathbb{F}_p$ where p is a prime different from 2 or 3.

As an example, consider the curve

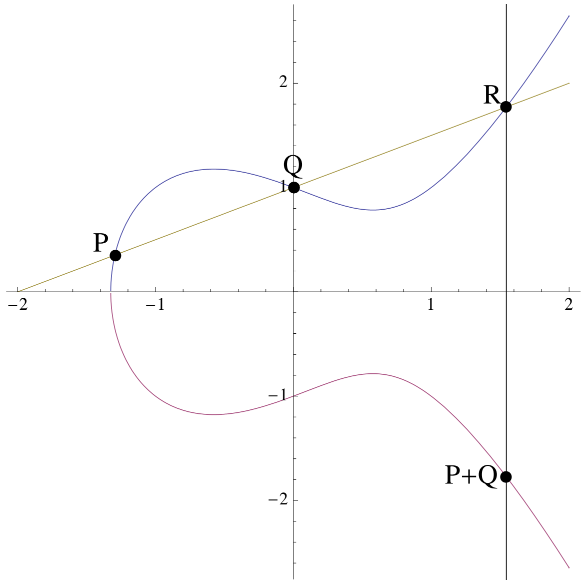
$$E_{-2,1} = \{(x, y) \in \mathbb{F}_7^2 : y^2 = x^3 - 2x + 1\} \tag{2}$$

over $\mathbb{F}_7$. This curve has $4a^3 + 27b^2 = -32 + 27 = -5 = 2 \bmod 7$, so it is nonsingular. It is tedious but straightforward to check that the points on this curve are

$$E_{-2,1} = \{\mathcal{O}, (0,1), (0,6), (1,0), (3,1), (3,6), (4,1), (4,6), (5,2), (5,5), (6,3), (6,4)\}. \tag{3}$$

In general, the number of points on the curve depends on the parameters a and b. However, for large p it is quite close to p for all curves. Specifically, a theorem of Hasse says it is $p+1-t$, where $|t| \le 2\sqrt{p}$. (Note that for elliptic curves, there is a classical algorithm, *Schoof's algorithm*, that computes the number of points on the curve in time $\mathrm{poly}(\log p)$. For more general curves defined by polynomial equations over finite fields, there are similar estimates to the one provided by Hasse's theorem, yet computing the precise number of points may be a classically hard problem. But for some such curves, there is an efficient *quantum* algorithm, *Kedlaya's algorithm*, for counting the number of points on the curve.)

It turns out that an elliptic curve defines an abelian group. Specifically, there is a binary operation '+' that maps a pair of points on the curve to a new point on the curve, in a way that satisfies all the group axioms. To motivate this definition, we go back to the case where $\mathbb{F} = \mathbb{R}$. Given two points $P, Q \in E_{a,b}$, their sum $P+Q$ is defined geometrically, as follows. For now, assume that neither point is $\mathcal{O}$. Draw a line through the points P and Q (or, if $P = Q$, draw the tangent to the curve at P), and let R denote the third point of intersection (defined to be $\mathcal{O}$ if the line is vertical). Then $P+Q$ is defined as the reflection of R about the x axis (where the reflection of $\mathcal{O}$ is $\mathcal{O}$). If one of P or Q is $\mathcal{O}$, we draw a vertical line through the other point, giving the result that $P + \mathcal{O} = P$: $\mathcal{O}$ acts as the additive identity. Thus we define $\mathcal{O} + \mathcal{O} = \mathcal{O}$. Note that reflection about the x axis corresponds to negation, so we can think of the rule as saying that the three points of intersection of any line with the curve sum to 0.



It turns out that this law makes $E_{a,b}$ into an abelian group for which the identity is $\mathcal{O}$ and the inverse of $P = (x,y)$ is $-P = (x,-y)$. By definition, it is clear that $(E_{a,b}, +)$ is abelian (the line through P and Q does not depend on which point is chosen first) and closed (we always choose $P+Q$ to be some point on the curve). The only remaining group axiom to check is associativity: we must show that $(P+Q)+T = P+(Q+T)$. Using a diagram of a typical curve, and picking three arbitrary points, you should be able to convince yourself that associativity appears to hold. Actually proving it in these geometric terms requires a little algebraic geometry.

For calculations, it is helpful to produce an algebraic description of the definition of elliptic curve point addition. Let $P = (x_P, y_P)$ and $Q = (x_Q, y_Q)$. The slope of the line through P and Q (with $P \neq Q$) is

$$\lambda = \frac{y_Q - y_P}{x_Q - x_P}. \tag{4}$$

Thus the set of points (x, y) on this line is $y = \lambda x + y_0$, where $y_0 = y_P - \lambda x_P$. Substituting this into (1) gives the equation

$$x^3 - \lambda^2 x^2 + (a - 2\lambda y_0)x + b - \lambda_0^2 = 0, \tag{5}$$

and solving this equation with the cubic formula shows that $x_P + x_Q + x_R = \lambda^2$. Thus we have

$$\begin{aligned} x_{P+Q} &= x_R \tag{6}\\ &= \lambda^2 - x_P - x_Q \tag{7}\\ &= \left(\frac{y_Q - y_P}{x_Q - x_P}\right)^2 - x_P - x_Q \tag{8} \end{aligned}$$

and

$$\begin{aligned} y_{P+Q} &= -y_R \tag{9}\\ &= -\lambda x_{P+Q} - y_0 \tag{10}\\ &= \lambda(x_P - x_{P+Q}) - y_P \tag{11}\\ &= \frac{y_Q - y_P}{x_Q - x_P}(x_P - x_{P+Q}) - y_P. \tag{12} \end{aligned}$$

A similar formula can be derived for the case where $P = Q$ (i.e., we are computing $2P$). It is straightforward to compute the slope of the tangent to the curve at P; if $y_P = 0$ then the slope is infinite, so $2P = \mathcal{O}$, but otherwise

$$\lambda = \frac{3x_P^2 + a}{2y_P}. \tag{13}$$

The rest of the calculation proceeds as before, and we have

$$\begin{aligned} x_{2P} &= \lambda^2 - 2x_P \tag{14}\\ &= \left(\frac{3x_P^2 + a}{2y_P}\right)^2 - 2x_P \tag{15} \end{aligned}$$

and

$$\begin{aligned} y_{2P} &= \lambda(x_P - x_{2P}) - y_P \tag{16}\\ &= \frac{3x_P^2 + a}{2y_P}(x_P - x_{2P}) - y_P. \tag{17} \end{aligned}$$

While the geometric picture does not necessarily make sense for the case where $\mathbb{F}$ is a finite field, we can take its algebraic description as a definition of the $+$ operation. Even over a finite field, it turns out that this operation defines an abelian group. It is a nice exercise to check explicitly (say, using Mathematica) that when addition of points is defined by these algebraic expressions, it is commutative, closed, and associative, thereby proving that $(E_{a,b}, +)$ is an abelian group over any field.

Elliptic curve cryptography Suppose we fix an elliptic curve $E_{a,b}$ and choose a point $g \in E_{a,b}$. Then we can consider the subgroup $\langle g \rangle$ (which is possibly the entire group if it happens to be cyclic). Using exponentiation in this group (which is simply multiplication in our additive notation), we can define analogs of Diffie-Hellman key exchange and related cryptosystems such as ElGamal. The security of such a cryptosystem then relies on the assumption that the discrete log problem on $\langle g \rangle$ is hard.

In practice, there are many details to consider when choosing an elliptic curve for cryptographic purposes. Algorithms are known for calculating discrete log on "supersingular" and "anomolous" curves that run faster than algorithms for the general case, so such curves should be avoided. Also, g should be chosen to be a point of high order—ideally, the elliptic curve group should be cyclic, and g should be a generator. Such curves can be found efficiently, and in the general case, it is not known how to solve the discrete log problem over an elliptic curve classically any faster than by general methods (e.g., Pollard's rho algorithm), which run in time $O(\sqrt{p})$.

Shor's algorithm for discrete log over elliptic curves It is straightforward to use Shor's algorithm to solve the discrete log problem for an elliptic curve over $\mathbb{F}_p$ in time $\operatorname{poly}(\log p)$. Points on the curve can be represented uniqely by their coordinates, with a special symbol used to denote $\mathcal{O}$, the point at infinity. Addition of points on the curve can be computed using the formulas described above, which involve only elementary arithmetic operations in the field. The most complex of these operations is the calculation of modular inverses, which can easily be done using the extended Euclidean algorithm.

Elliptic curve cryptosystems are commonly viewed as being more secure than RSA for a given key size, since the best classical algorithms for factoring run faster than the best classical algorithms for calculating discrete log in an elliptic curve. Thus, in practice, much smaller key sizes are used in elliptic curve cryptography than in factoring-based cryptography. Ironically, Shor's algorithm takes a comparable number of steps for both factoring and discrete log (regardless of the group involved), with the caveat that group operations on an elliptic curve take more time to calculate than ordinary multiplication of integers.