University of New Brunswick

Honors project

Applications of Elliptic Curves Over Finite Fields

Author:
Josh Koncovy

Supervisor:
Dr. Colin Ingalls

April 29, 2014

Elliptic curves

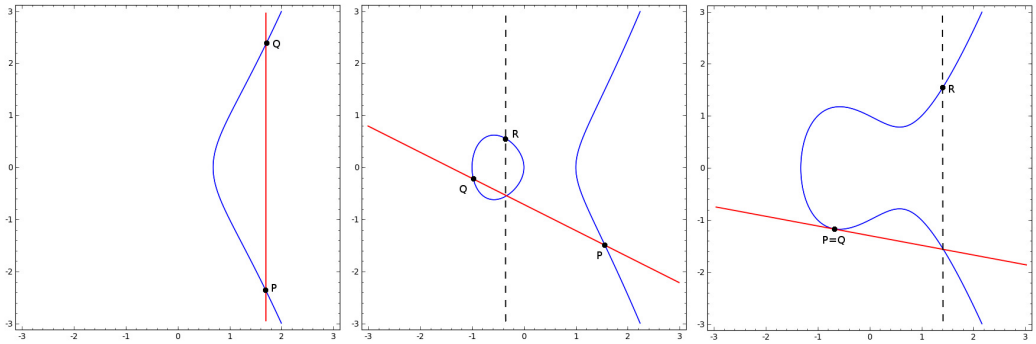
Elliptic curves are generally expressed as Weierstrass equations, Let $\mathbf{F}$ be a field

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

where $a, b \in \mathbf{F}$. As long as the field is not characteristic 2 or 3, and the curve is non-singular. Non-singularity gives the desirable geometric properties of no cusps or self intersections, it means algebraically that the discriminant $\Delta = -16(4a^3 + 27b^2)$ must be non zero. So these curves can be viewed as taking a variable squared and setting equal to any cubic polynomial with distinct roots.

Defining groups with elliptic curves

These curves can be used to form Abelian groups, $E(\mathbf{F})$ is the set of all points in $\mathbf{F}$ that satisfy the equation together with an additional point I at infinity. The point at infinity gives the projective closure of the curve and acts as the group identity. The group operation is defined by taking the unique collinear point that intersects the curve and then inverting it about the x-axis. If the two points are equal the tangent to that point is used to find the resulting point. The following graphic shows geometrical examples of arithmetic on some elliptic curves



When two points on the curve P and Q have the same x value and negative y values of each other then R = I as shown on the left.
If P = Q we use the tangent line to P to find the intersection then invert to obtain R as in the rightmost example.
The fact this operation does indeed form an Abelian group comes from the various geometrical properties of theses curves. Inverses are obtained from the symmetry about the x-axis, commutativity from the adjoining line between two points, and so forth. The only non obvious property is associativity which can be proved tediously case by case, a geometrical proof can be found

in W. Fulton's Algebraic curves.
The group law can also be defined algebraically by the following, given an elliptic curve $y^2 = x^3 + ax + b$ and points $P = (x_P, y_P)$ and $Q = (x_Q, y_Q)$

$$P + Q = R = (x_R, y_R)$$

If $x_P = x_Q$ and $y_P = -y_Q$ then $R = I$ Otherwise

$$x_R = \lambda^2 - x_P - x_Q$$
$$y_R = \lambda(x_P - x_R) - y_P$$

where λ is defined by

$$\lambda = \begin{cases} (y_Q - y_P)(x_Q - x_P)^{-1} : x_P \neq x_Q \\ (3x_P^2 + a)(2y_P)^{-1} : x_P = x_Q, y_P = y_Q \end{cases}$$

Here λ is the slope of the line calculated using grade school methods if P is not equal to Q or -Q, and using a derivative else wise.

Elliptic curves over finite fields

Over finite fields, elliptic curves yield finite Abelian groups. For example the curve $y^2 = x^3 + 3x + 2$ over $\mathbf{F}_{11}$ has the corresponding group

$$\{(2, \pm 4), (3, \pm 4), (4, \pm 1), (6, \pm 4), (7, \pm 5), (10, \pm 3), I\}$$

Since the order of the group is 13 any non-identity element acts as a generator. The exact size of these groups is open to conjecture but Hasse's theorem on elliptic curves provides a bound to give some idea, if N is the number of points that satisfy some elliptic curve over $\mathbf{F}_q$ then

$$|N - (q + 1)| \leq 2\sqrt{q}$$

Schoof's algorithm gives a method of counting points on Elliptic curves in polynomial time making use of Hasse's bound as well as the Chinese remainder theorem and division polynomials. The algorithm can be found in Schoof's Counting Points on Elliptic Curves over Finite Fields.
Elliptic curves over finite fields have applications in a number of algorithms including cryptography and integer factorization.

Elliptic curve cryptography

These groups can be used to perform public key cryptography that utilizes their algebraic structure. In particular, it is easy to compute powers of some element, but hard to take logarithms. Several algorithms have been made to perform this task, a simple way to construct such schemes is to take already existing protocols using $(\mathbf{Z}_p)^\times$ as the active group and replacing with a group from an elliptic curve, common examples of such are elliptic curve Diffie-Hellman and Elliptic Curve Digital Signature Algorithm. These procedures have recently achieved popularity due to the difficulty of the discrete logarithm problem in comparison to number fields, which allows choice of smaller key sizes then there integer counterparts. In general a curve and a base point are decided on and made public, the selection of these parameters depends on the key size, commonly used domain parameters as well as some rhetoric on selection that can be found in a publication by NIST.

ElGamal encryption

An example of elliptic curve cryptography can be constructed using a general method and applying to elliptic curves. Consider ElGamal encryption, a method of public key cryptography, which can be defined over any cyclic group. The usage of the ElGamal cryptosystem will be shown through the following scenario. Suppose Bob wishes to send Alice a message

- A cyclic group G is decided on with order q and generator g
- Alice selects a number x between 0 and q, this is her private key
- Alice computes $h = g^x$ and makes it public
- Bob also chooses a number y between 0 and q
- Bob calculates the shared secret $s = h^y$
- Bob converts his message m into an element m' of the group G
- Bob computes $c_1 = g^y$ and $c_2 = m' * s$ and sends (c_1, c_2) to Alice
- Alice gets the shared secret by $s = c_1^x$
- Alice can now retrieve the message by using $m' = c_2 * s^{-1}$

A scenario of using ElGamal with an Elliptic curve group would go as follows.

- A curve is selected $y^2 = x^3 + 13x + 37\ mod(101)$ as well as a generator $g = (36, 87)$ and order is computed $q = 108$ and all are made public
- Alice selects a number $x = 41$ as her private key and then computes and publishes $h = g^x = 41(36, 87) = (48, 31)$
- Bob also chooses a number $y = 73$ and calculates the shared secret $s = h^y = 73(48, 31) = (60, 77)$
- Bob has the private message $m = (85, 51)$
- Bob computes $c_1 = g^y = 73(36, 87) = (49, 70)$ and $c_2 = m' * s = (85, 51) + (60, 77) = (66, 6)$ and sends $((49, 70), (66, 6))$ to Alice
- Alice now decrypts the message by getting the shared secret $s = c_1^x = 41(49, 70) = (60, 77)$ then $m = c_2 * s^{-1} = (66, 6) + (60, -77) = (85, 51)$

The selection of a message in the form of a group element can be done by some agreed upon hash-table taking elements of the group to hex-keys or letters.

Elliptic curve factorization

Elliptic curve algebra may also be used for integer factorization. The Lenstra elliptic curve factorization also known as elliptic curve factorization method (ECM) runs in sub exponential time and is the third fastest known method of integer factorization, behind only multiple polynomial quadratic sieve and the general number field sieve. The algorithm runs as follows, given some natural number n to factor:

1. Pick a random elliptic curve over $\mathbf{Z}/n$ and a non-trivial point $P = (x_0, y_0)$ satisfying the curve. In practice pick a point and some value for a then compute b.
2. Compute eP where e is a product of small numbers, commonly factorials are used i.e. compute $2P, 6P, (4!)P, (5!)P$ etc until computations become time intensive. Here since the curve is not over a field is possible to get division by zero in the calculation, if this happens the algorithm finishes.
3. When calculating eP one is required to find inverses to compute slope λ, in the previously mentioned equations. Typically the Euclidean

algorithm is used for this, and if an element v is found to be not invertible, then it is the case that $gcd(v, n) \neq 1, n$ and the algorithm has succeeded.

4. If no such elements are found or the identity arises then new parameters are to be selected and the algorithm is ran again.

The algorithm works on the following mechanics, suppose p and q are prime factors of n, then the curve over p and q are groups. In general this is the Chinese remainder theorem $\mathbf{Z}/n = \mathbf{Z}/p + \mathbf{Z}/q$ The order of the groups are random numbers near p+1 and q+1 respectively by Hasse's bound, which makes it unlikely that that the order of the groups share common prime factors, so one is likely to find some point that is the identity in one group but not the other. When this occurs a factor of only one of the groups is obtained and thus a non trivial factor of n. A detailed discussion the running time can be found in Factoring Integers with Elliptic Curves by Lenstra

For example suppose we wish to factor n = 38911
Firstly choose a point and a value say P = (1,1) and a = 1, then find a curve by solving for b

$$1^2 = 1^3 + 1 + b \; mod(38911)$$

so b = -1 and the curve is

$$y^2 = x^3 + x - 1 \; mod(38911)$$

Then compute appropriate eP,

$$2P = (1, 1) + (1, 1) = (2, 38908)$$

To show how this algorithm finds factors this computation requires inverting 2 in $\mathbf{Z}/38911$. This is done using the extended Euclidean algorithm $gcd(2, 38911) = 1 = 38911(1) + 2(-19455)$ so $2^{-1} = -19455 \; mod(38911)$
Continue computing eP until $(6!)P$, this computation requires inverting 29392 but $gcd(29392, 38911) = 167$ so we have found a factor of n and now know $38911 = 167 * 233$

Elliptic curve primality proving

Elliptic curves can also be used to determine the primality of numbers, mainly from the following proposition

Proposition

Let N be a natural number and E be some elliptic curve defined over $\mathbf{Z}/n$. Let m be an integer. If some prime $q < (N^{1/4} + 1)^2$ divides m and there exists a point P on E such that

1. $mP = 0$
2. $(m/q) \neq 0$ and is defined

Then N is prime

Proof

Suppose N is a composite number, this implies there is a prime factor $p \leq \sqrt{N}$. Now consider E_p the group formed by taking $E\ mod(p)$ as explained above. Let m be the number of points that satisfy E. Using Hasse's bound

$$m \leq p + 1 + 2\sqrt{p} = (\sqrt{p} + 1)^2 \leq (N^{1/4} + 1)^2 < q$$

using this and the primality of q $gcd(q, m) = 1$ and thus from Bzout there exists an integer u such that $u * q = 1\ mod(m)$. so in E_p

$$(m/q) = uq(m/q)P = umP = 0$$

since the first property of P in the proposition gives $mP = 0$ in E it must also be true in E_p, as p divides N. Note that here P is taken modulo p not N.
Since $(m/q)P = 0$ in E_p it is also true in E which gives a contradiction of the second property of P, and thus the proposition is proven.

Goldwasswe-Kiilian algorithm

The Goldwasser-Kilian algorithm provides a means of implementing the above proposition to determine primality. Similarly to Lenstra factorization one selects an elliptic curve E over N and some point on it P. Next the number of points on E is computed. Then a criterion for deciding weather or not the curve is acceptable is used. If the number of points on the curve can be expressed as $m = kq$ where, k is a small integer greater than one and q is probably prime, then the curve is good. If not a new curve and point are selected. Next mP and kP are calculated. If an undefined expression arises then a factor is found and the number is not prime. If $mP \neq 0$ then N is composite, since if N is prime E_N has order m and multiplying m by any

element must be 0. If $kP = 0$ then the algorithm must be ran again with a new curve and point. If neither of these are true then $mP = 0$ and $kP \neq 0$ then by the proposition N is prime.
This method gives a smaller number q which now has to be checked for true primality, done recursively with this algorithm until q is small enough to check using slower methods.

Atkin-Morain elliptic curve primality test

A much more efficient method of testing primality with the proposition was provided by Atkin and Morain. The Atkin-Morain elliptic curve primality test or just ECPP addresses the problem of having to count points using Schoof or other methods, by selecting particular curves such that the number of points is easy to count. Explanation of this can be found in Atkin and Morain's Elliptic Curves and Primality Proving. ECPP is of particular note as it is currently the fastest in use primality test for general numbers (not of a special form) as claimed by Lenstra in Algorithms in number theory. It runs heuristically in $O((logn)^{5+\epsilon})$ for some $\epsilon > 0$.

References and further reading

The Arithmetic Of Elliptic Curves, 2nd Edition, Joseph H. Silverman
Algebraic Curves, An Introduction to Algebraic Geometry, William Fulton
Counting Points on Elliptic Curves over Finite Fields, R. Schoof
Recommended Elliptic Curves For Federal Government Use, July 1999, NIST
Factoring Integers With Elliptic Curves, The Annals of Mathematics, H. W. Lenstra Jr.
Elliptic Curves and Primality Proving A. O. L. Atkin, F. Morain