Elliptic Curves and Cryptography

David A. Craven

3rd February 2010

1 Introduction and Terminology

Generally, we have the following terminology and scenario:

- *Alice* has a message, that she wants to send to *Bob*. *Eve* would like to know it. Depending on the scenario, Eve has access to various methods, including being able to listen in on everything sent between Alice and Bob, being able to forge data to look as though it comes from Alice and send it to Bob, and so on.
- The message itself is often thought of as a number, being translated into such probably by something like ASCII. It is called the *plaintext*.
- A method of turning plaintext into something else is called a *cypher*. The cypher normally takes two inputs: the plaintext and a *key*, which will be used to obfuscate the message. The key in general should be private, so that only Alice and Bob know the key. The result of applying the cypher is the *cyphertext*.
- The cyphertext is transmitted over open channels, and Eve has access to that.
- Traditionally, the key to encode a message was the same as the key to decode the message. (One method was to simply add the key and the plaintext together, both being thought of as points in $\mathbb{F}_2^n$.) In *public-key cryptography*, the key to encode the message is *not* the same as the key to decode it. The key to encode the message is broadcast in open channels, and can be assumed to be known by Eve.

2 Public Key Cryptography

The RSA algorithm was developed by Cliff Cocks in 1973. Basically, it works as follows:

(i) Choose two (large) prime numbers p and q, and computer $n = pq$.

(ii) Choose an integer e with $1 < e < \phi(pq)$, such that e and $\phi(pq)$ are coprime. (Normally, choose $e = 2^{16} + 1$ because it is easy to multiply up.)

(iii) Let $d = e^{-1}$ in the ring $\mathbb{Z}_{(p-1)(q-1)}$.

(iv) Tell everyone n and e.

Now suppose that Alice has a message, m. The cyphertext c is given by $c = m^e \bmod n$. She sends c to Bob, who secretly knows d. Bob computes $c^d \bmod n$, and gets c. Brilliant!

Let us prove that this works though. By Euler's theorem, $m^{\phi(n)} \equiv 1 \bmod m$, and so for any $\alpha \in \mathbb{Z}$, we have $m^{\alpha\phi(n)+1} \equiv m \bmod n$. Since e is prime to $\phi(n)$, if d is the inverse of e in the ring $\mathbb{Z}_{\phi(n)}$, then we have $de = \alpha\phi(n) + 1$ for some $\alpha \in \mathbb{Z}$. Hence

$$m^{de} \equiv m^{\alpha\phi(n)+1} \equiv m \mod n,$$

proving the claim.

Clearly if you know p and q, computing d is easy using the Euclidean algorithm. We want to know that computing d from n and e is *equivalent* to knowing p and q. Therefore, we assume that we know d, e and n, and need to compute p and q. Let s be the largest number such that $2^s \mid (ed-1)$, and write $k = (ed-1)/2^s$.

We first prove that for all integers a prime to n, the order of a^k in $\mathbb{Z}_n$ divides 2^s. Since $a^{de-1} \equiv 1 \bmod n$ (as RSA works) and $ed - 1 = k2^s$, we have $(a^k)^{2^s} \equiv 1 \mod n$; hence a^k has order a divisor of 2^s.

Now the idea is to choose an integer at random between 1 and $n-1$, and compute the gcd d between it and n. If $d = 1$ then compute $d = \gcd(a^{2^i k} \bmod n, n)$ for $i = s-1, s-2$ and so on until either $d > 1$ or $s = 0$. If $d > 1$ then we are done! Otherwise, we fouled up.

Theorem 2.1 This algorithm fouls up at most half the time.

The reason for this is as follows: if a has a different order modulo p and mod q, then $\gcd(a^{2^i k} - 1, n)$ is either p or q for some $0 \leqslant i \leqslant s-1$. The last piece of information that you need is that there are at least $(p-1)(q-1)/2$ numbers a for which a^k has a different order modulo p and q.

3 Pollard's $p-1$ Algorithm and Strong Primes

Now we have RSA, let's try to break it. We need to come up with a way of factorizing numbers. One example is Pollard's $p-1$ algorithm.

(i) Choose a bound B, and (probably small) a prime to n.

(ii) Let k be the product of all prime powers less than B.

(iii) Let d be $\gcd(a^k - 1, n)$.

(iv) If $d = 1$, then B was too small. If $d = n$ then B was too big!

How is this going to find anything at all? Define a number x to be *B-powersmooth* if all prime powers dividing x are at most B. The gcd d above selects all primes p dividing n such that $p-1$ is B-powersmooth. To see this, suppose that $p \mid n$, and p is B-powersmooth. Then k is a multiple of $p-1$, and so by Fermat's little theorem $a^k \equiv 1 \bmod p$. In particular, this means that p divides $a^k - 1$, and so divides d.

Similarly, there is Williams's $p+1$ method, that finds prime divisors p of n such that $p+1$ is B-powersmooth. Therefore, we make the following definition.

Definition 3.1 A large prime p is said to be *strong* if both $p-1$ and $p+1$ have at least one large prime divisor.

It should be pointed out that most randomly generated large primes are strong, so in practice, unless someone deliberately cripples you, everything should be fine regarding this particular algorithm. But not the next one, which will need a section on elliptic curves first.

4 Elliptic Curves

Let E be an elliptic curve over $\mathbb{R}$. Without loss of generality, we may assume that E is given by $y^2 = x^3 + ax + b$. The set of points on E can be given a groups structure, well, *almost*. Given two points P and Q on an elliptic curve, we draw the line connecting them. This will intersect the curve in another point, probably. If it does, we want to define $P + Q$ as the intersection, but that isn't quite right. If this point is (x, y), then we finally reflect in the x-axis to get $(x, -y)$. This is the point $P + Q$. The only cases we haven't defined it is when the line PQ only intersects P and Q, and when $P = Q$. In the latter case we simply use the tangent, and in the former case, the sum is defined to be ∞, i.e., the point at infinity.

The result is that this makes $E(\mathbb{R})$, the set of points of E over $\mathbb{R}$, into an abelian group, if ∞ is thrown into the mix. In order to work over other fields, it seems clear that we need to make this geometric construction algebraic.

Let (x_1, y_1) and (x_2, y_2) denote the co-ordinates of P and Q. Suppose that $x_1 \neq x_2$, so that we are in the general case. The line connecting P and Q is given by the formula

$$\frac{y - y_1}{y_2 - y_1} = \frac{x - x_1}{x_2 - x_1}.$$

If $\alpha = (y_2 - y_1)/(x_2 - x_1)$ and $\beta = y_1 - \alpha x_1$, then this becomes $y = \alpha x + \beta$. A point $(x, \alpha x + \beta)$ lies on E if and only if $(\alpha x + \beta)^2 = x^3 + ax + b$, and we get the cubic

$$x^3 - \alpha^2 x^2 + (a - 2\alpha\beta)x + (b - \beta^2).$$

This equation has two solutions already, namely x_1 and x_2, and so must have exactly one more, $x_3 = \alpha^2 - x_1 - x_2$. (This comes from the fact that the trace is $-\alpha^2$.) This therefore gives the co-ordinates of the final point, reflected, as

$$\left(\alpha^2 - x_1 - x_2, -(\alpha(\alpha^2 - x_1 - x_2) + \beta)\right).$$

If $P = Q$, which we will have to deal with, we have that $\alpha = \mathrm{d}y/\mathrm{d}x$, so we get (if the co-ordinates of $2P$ are (x_3, y_3)),

$$\alpha = \frac{3x_1^2 + a}{2y_1}, \quad x_3 = \left(\frac{3x_1^2 + a}{2y_1}\right)^2 - 2x_1, \quad y_3 = -y_1 + \left(\frac{3x_1^2 + a}{2y_1}\right)(x_1 - x_3).$$

This algebraic description makes it easy to translate the result into other fields, since all we are doing is using field operations. Hence there are abelian group structures on $E(\mathbb{F})$ for any field $\mathbb{F}$.

The last piece of information that we need is Hasse's theorem on $|E(\mathbb{F}_q)|$.

Theorem 4.1 (Hasse) Let E be an elliptic curve over a field $\mathbb{F}_q$. Then

$$q + 1 - 2\sqrt{q} \leqslant |E(\mathbb{F}_q)| \leqslant q + 1 + 2\sqrt{q}.$$

For different choices of a and b we get different sizes of $|E(\mathbb{F}_q)|$ for a given prime power q.

If R is a ring, rather than a field, then we can define $E(R)$ perfectly well. This fails to inherit an abelian group structure, although some products are defined. The only thing that we need to invert is the denominator of the slope α of the line. If $x_2 - x_1$ or $2y_1$ is invertible in R, then this addition can be performed in $E(R)$. This partial abelian group structure is important, notably because it fails in $R = \mathbb{Z}_n$ if and only if the denominator and n have a common factor!

5 Lenstra's Elliptic Curve Factorization

This is the third-fastest factoring algorithm, according to Wikipedia, behind two number field sieve algorithms, that I will not talk about.

The idea is that the number of points on an elliptic curve over $\mathbb{F}_p$ is somewhere between $p+1-2\sqrt{p}$ and $p+1+2\sqrt{p}$. Whereas the $p-1$ algorithm failed if $p-1$ is not powersmooth, this algorithm fails if $|E(\mathbb{F}_p)|$ is not powersmooth. Since this bops around randomly in the interval above, we stand a good (subexponential) chance that it might well be powersmooth, and then we are in business.

We have a number n to factorize, as always.

(i) Let E be an elliptic curve over $\mathbb{Z}_n$, say $y^2 = x^3 + ax + b$, and let $P = (x, y)$ be a non-trivial point on E.

(ii) Try to compute $B! \cdot P$ for small numbers B. One of three things might happen.

  (a) It works, and you don't get ∞. In this case, B was too small.

  (b) It works, and you get ∞. In this case, B was too large, and you have to start again.

  (c) At some point you cannot multiply the points, because you are on an elliptic curve over $\mathbb{Z}_n$ rather than over a field, and you cannot invert the denominator of the slope.

(iii) If (a) happens, choose a different curve and start again. If (b) happens, you have the right curve, choose a smaller B and/or another point P. If (c) happens, party time.

Why does this work at all efficiently? E is an elliptic curve over $\mathbb{Z}_n$; if $p \mid n$ then we can think about $E(\mathbb{F}_p)$, and every point on $E(\mathbb{Z}_n)$ projects onto $E(\mathbb{F}_p)$. The number of points $|E(\mathbb{F}_p)|$ is approximately $p+1$, but since it varies a little, it could be that $|E(\mathbb{F}_p)|$ is smooth, or something similar, like $|E(\mathbb{F}_p)| \mid B!$ for some 'small' B.

Suppose that $|E(\mathbb{F}_p)| \mid B!$. When computing $B! \cdot P$, we get that the slope of the line has denominator 0. However, if this *isn't* true for one of the other factors q dividing n we will get something non-zero in the specialization $E(\mathbb{F}_q)$. This means that the point *doesn't exist* in $E(\mathbb{Z}_n)$, and so the multiplication breaks down.

This algorithm actually works fairly well, and is also really cool.

6 Elliptic Curve Cryptography

Having used elliptic curves to try to break cyphers, let's use elliptic curves to try to *make* cyphers. One thing we want to do is produce a private key that two people can use to communicate, but be able to do this in the open. The standard method is to use the discrete-log problem.

Let p be a prime number. If I give you $a \bmod p$ and $a^k \bmod p$, can you tell me $k \bmod (p-1)$? The answer is yes, obviously, but with great difficulty. This is the basis of the standard Diffie–Hellman key exchange.

Alice sends Bob a prime p and a base a. Both Alice and Bob choose private numbers k and l. Alice sends Bob a^k and Bob sends Alice a^l. Then both raise their received numbers to the apprropriate powers to get a^{kl}. Eve knows p, a, a^k, and a^l, but not k or l. The problem is to construct a^{kl} from these data. At the moment it is not known, but suspected, that obtaining a^{kl} is equivalent to getting either k or l.

Like public-key cryptography, there are several sub-exponential-time algorithms to solve the discrete-log problem. To Koblitz and Miller, this was not helpful, and so they (independently) suggested using elliptic curves to make things even harder.

In this implementation, Alice decides on an elliptic curve E over a field $\mathbb{F}_p$, and a random point P. Again Alice dreams up a number k and Bob a number l, and they send each other kP and lP, then multiply the public point by their private multiplier to get the answer. Simples.

Even in this case you can do something, but not very often. One method is to try to reduce the problem to the standarrd discrete log by using the Weil pairing (this embeds $E(\mathbb{F}_p)$ insude $\mathbb{F}_{p^a}$ for some a, in a certain sense). This doesn't work, however, unless the Weil pairing produces a low a, which *doesn't happen*.