Chapter 5.6

Lenstra's elliptic curve factorization algorithm

Recall Pollard's $p-1$ factorization method, which finds factors of $N = pq$ by searching for a power a^L w/ the property that

$$a^L \equiv 1 \pmod{p} \quad \text{and} \quad a^L \not\equiv 1 \pmod{q}$$

By Fermat's Little theorem, this is likely to work if $p-1$ divides L and $q-1$ does not divide L. So we take $L = n!$ for some moderate value of n.

Then we hope that $p-1$ or $q-1$ (but not both) is a product of small primes, hence divides $n!$

Pollard's method works well for some numbers, but not all.
← the determining factor is whether $p-1$ or $q-1$ is a product of small primes

- The important part of this is $p-1$. there are $p-1$ elements in $\mathbb{F}_p^*$, so every element α of $\mathbb{F}_p^*$ satisfies $\alpha^{p-1} = 1$

* - For elliptic curves, this last statement is analogus to them because the points and the addition law for an Elliptic curve $E(\mathbb{F}_p)$ work similarly to the elements and the multiplication law for $\mathbb{F}_p^*$ *

Hendrik Lenstra made this analogy precise by creating a factorization algorithm that uses the group law on an elliptic curve E in place of multiplication modulo N.

• To describe this algorithm, we will ~~start~~ work w/ an elliptic curve modulo N, where integer N is not prime, so the ring $\mathbb{Z}/N\mathbb{Z}$ is not a field

Suppose we have the curve

$$E: Y^2 = X^3 + AX + B$$

and suppose that $P = (a, b)$ is a point on E modulo N, which means:

$$b^2 \equiv a^3 + A \cdot a + B \pmod{N}$$

Then we can apply the EC addition algorithm to compute $2P, 3P, 4P, \ldots$, since the only operations required by that alg. are add, sub, mult, and div (by # relatively prime to N)

Ex. Let $N = 187$ and consider the EC

$$E: Y^2 = X^3 + 3X + 7 \pmod{187}$$

and the point $P = (38, 112)$, that is on E modulo 187.
We compute $2P$ by the addition alg.

* Note: $x(P)$ and $y(P)$ stand for x, y coord. of P & similarly for $2P$

we compute this by EEA $\quad \frac{1}{2y(P)} = \frac{1}{224} \equiv 91 \pmod{187}$

$$\lambda = \frac{3x(P)^2 + A}{2y(P)} = \frac{4335}{224} \equiv 34 \cdot 91 \equiv 102 \pmod{187}$$

$$x(2P) = \lambda^2 - 2x(P) = 10328 \equiv 43 \pmod{187}$$

$$y(2P) = \lambda(x(P) - x(2P)) - y(P) \equiv 102(38 - 43) - 112 \equiv 126 \pmod{187}$$

Thus $2P = (43, 126)$ as a point on E modulo 187

Now, we compute $3P = 2P + P$ in a similar method (we're adding distinct points, so the formula for λ is slightly different)

$$\frac{1}{x(2P) - x(P)} = \frac{1}{5} \equiv 75 \pmod{187}$$

$$\lambda = \frac{y(2P) - y(P)}{x(2P) - x(P)} = \frac{14}{5} \equiv 14 \cdot 75 \equiv 115 \pmod{187}$$

$$x(3P) = \lambda^2 - x(2P) - x(P) = 13144 \equiv 54 \pmod{187}$$
$$y(3P) = \lambda^*(x(P) - x(3P)) - y(P) = 115(38 - 54) - 112 \equiv 105 \pmod{187}$$

Thus $3P = (54, 105)$

Now let's attempt to calculate $5P$ for $P = (38, 112)$ on the same curve

$$5P = 3P + 2P$$

$$x(3P) - x(2P) = 54 - 43 = 11 \pmod{187}$$

because the $\gcd(11, 187) > 1$ it gives use a divisor of 187. So now we can factor 187

$$187 = 11 \cdot 17$$

This idea is the underlying property for Lenstra's EC factorization alg.

If we look at the curve $E \bmod 11$,

$$P = (38, 112) \equiv (5, 2) \bmod 11$$

satisfies $5P = \mathcal{O}$ in $E(\mathbb{F}_{11})$ so this means at some stage of the calculation we tried to divide by 0. but 0 here is in $\mathbb{F}_{11}$, so we actualy end up trying to find a the reciprocal modulo 11 of some integer that is divisible by 11

So from this example, we replace multiplication modulo N in Pollard's factorization method with EC addition modulo N.

- We start with an elliptic curve E and a point P on $E \bmod N$ and compute

$$2!P,\ 3!P,\ 4!P,\ 5!P,\ \ldots \pmod N$$

once we have computed $Q = (n-1)! \cdot P$ it is easy to compute $n!P$ since it equals nQ.

~~The alg~~ At each stage, 3 things may occur.

1) we are able to compute $n!$
2) we may need to find the reciprocal of a number d that is a multiple of N, which is unlikely
3) we may need to find the reciprocal of a number d that satisfies $1 < \gcd(d, N) < N$, in which case $n! \cdot P$ fails but $\gcd(d, N)$ is a non trivial factor of N

Lenstra's Algorithm

Input: Integer N to be factored

1. Choose random values A, a, and b modulo N.
2. Set $P = (a, b)$ and $B \equiv b^2 - a^3 - A \cdot a \pmod{N}$
   Let E be the ec $E: Y^2 = X^3 + AX + B$
3. Loop $j = 2, 3, 4, \ldots$ up to specified bound
   4. Compute $Q \equiv jP \pmod{N}$ and set $P = Q$
   5. If (4) fails, then we have a $d > 1$ with $d \mid N$
   6. If $d < N$, success, return d
   7. If $d = N$, goto (1) & choose a new curve and point
8. increment j and loop again @ (2)

- Initial problem: finding a P on E mod N - fix by constructing ec from P

Ex. $N = 6887$,
randomly select $P = (1512, 3166)$ and $A = 14$

so $B \equiv 3166^2 - 1512^3 - 14 \cdot 1512 \equiv 19 \pmod{6887}$

so $E: Y^2 = X^3 + 14X + 19$

Now we compute multiples of P:

so $2P \equiv (3466, 2996) \pmod{6887}$

Next, $3! \cdot P = 3(2P) = 3 \cdot (3466, 2996) \equiv (3067, 396) \pmod{6887}$

table

n	$n! \cdot P \bmod 6887$
1	$P =$ (1512, 3166)
2	$2! \cdot P =$ (3466, 2996)
3	$3! \cdot P =$ (3067, 396)
4	$4! \cdot P =$ (6507, 2654)
5	$5! \cdot P =$ (2783, 6278)
6	$6! \cdot P =$ (6141, 5581)

These values in the table aren't interesting. However, when we try to compute $7!P$, we have

$$\begin{aligned} 7Q &\equiv (Q + 2Q) + 4Q \pmod{6687} \\ &\equiv ((6141, 5581) + (5380, 174)) + (203, 2038) \\ &\equiv (984, 589) + (203, 2038) \end{aligned}$$

to perform the final step, we need to compute the reciprocal of $203 - 984 \bmod 6887$, but we find that

$$\gcd(203 - 984, 6887) = \gcd(-781, 6887) = 71$$

so now we have the factorization of 6887, 71, so

$$6887 = 71 \cdot 97$$

time complexity:

$O(e^{\sqrt{2(\log p)(\log\log p)}})$ steps

if p is the smallest factor of N

Ask about "sieve factorization"

Chapter 5.7

Elliptic Curves over $\mathbb{F}_2$ and over $\mathbb{F}_{2^k}$

- It is suggested that it would be efficient to use elliptic curves modulo 2, since binary is the language of computers.
- However, if we define an elliptic curve over $\mathbb{F}_2$, then $E(\mathbb{F}_2)$ contains at most 5 points, so $E(\mathbb{F}_2)$ is not useful for cryptography

→ But we have other finite fields in which $2 = 0$. These are the fields $\mathbb{F}_{2^k}$ containing 2^k elements. Recall for every prime power p^k there exists a field $\mathbb{F}_{p^k}$ with p^k elements. So we can take an elliptic curve whose Weierstrass equation has coeff in a field $\mathbb{F}_{p^k}$ and look at the group of points on that curve having coordinates in $\mathbb{F}_{p^k}$.

Hasse's theorem is true in this more general setting

Theorem (Hasse): Let E be an elliptic curve over $\mathbb{F}_{p^k}$. Then $\#E(\mathbb{F}_{p^k}) = p^k + 1 - t_{p^k}$ with t_{p^k} satisfying

$$|t_{p^k}| \leq 2p^{k/2}$$

Ex. $\mathbb{F}_4 = \{ a + bi \; ; \; a, b \in \mathbb{F}_3 \}$ where $i^2 = -1$

EC over $\mathbb{F}_9$,

$$E: Y^2 = X^3 + (1+i)X + (2+i)$$

There are 10 points in $E(\mathbb{F}_9)$

$\mathcal{O}$ $(2i, 1+2i)$ $(2i, 2+i)$ $(1+i, 1+i)$
$(1+i, 2+2i)$ $(2,0)$ $(2+i, i)$ $(2+i, 2i)$
$(2+2i, 1)$ $(2+2i, 2)$

The goal is to use $\mathbb{F}_{2^k}$ for cryptography, but there is one problem to be addressed.

The discriminant is actually defined as

$$\Delta = -16(4A^3 + 27B^2), \text{ not}$$
$$\Delta = 4A^3 + 27B^2 \neq 0$$

As long as we work in a field where $2 \neq 0$, then the condition $\Delta \neq 0$ is the same with either def., but for fields such as $\mathbb{F}_{2^k}$ where $2 = 0$, we have $\Delta = 0$ for every standard Weierstrass equation. The solution is to enlarge the collection of allowable Weierstrass equations

Def An elliptic curve E is the set of solutions to a generalized Weierstrass equation

$$E: Y^2 + a_1XY + a_3Y = X^3 + a_2X^2 + a_4X + a_6$$

together with an extra point $\mathcal{O}$

the coeff are required to satisfy $\Delta \neq 0$ where the discriminant Δ is defined in terms of certain quantities b_2, b_4, b_6, b_8

$$b_2 = a_1^2 + 4a_2, \quad b_4 = 2a_4 + a_1a_3, \quad b_6 = a_3^2 + 4a_6$$
$$b_8 = a_1^2a_6 + 4a_2a_6 - a_1a_3a_4 + a_2a_3^2 - a_4^2$$

$$\Delta = -b_2^2b_8 - 8b_4^3 - 27b_6^2 + 9b_2b_4b_6$$

The geometric definition of the addition law on E is now (for the reflection step on y)

$$(x, y) \longmapsto (x, -y - a_1x - a_3)$$

Working with generalized Weierstrass equations, we can derive an addition algorithm similar to the previous one.

ex. if $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$ where $P_1 \neq P_2$ the x-coord. of their sum is

$$x(P_1 + P_2) \equiv \lambda^2 + a_1\lambda - a_2 - x_1 - x_2 \quad \text{with } \lambda = \frac{y_2 - y_1}{x_2 - x_1}$$

and for duplicating a point P,

$$x(2P) = \frac{x^4 - b_4x^2 - 2b_6x - b_8}{4x^3 + b_2x^2 + 4b_4x + b_6}$$

- advantages to working with elliptic curves over $\mathbb{F}_{2^k}$ rather than $\mathbb{F}_q$
  - the binary nature of computers eases computations
  - take k as a composite number in which case $\mathbb{F}_{2^k}$ contains other finite fields intermediate b/t $\mathbb{F}_2$ and $\mathbb{F}_{2^k}$

faster computationally, but can also cause security problems. best option for security purposes is to take k as a prime

- greatest advantage is to use an elliptic curve E over $\mathbb{F}_2$ while taking points on E w/ coord. in $\mathbb{F}_{2^k}$
  - This allows for the use of the Frobenius map ← signif. gain in efficiency

Def: The (p-power) Frobenius map τ is the map from the field $\mathbb{F}_{p^k}$ to itself defined by

$$\tau : \mathbb{F}_{p^k} \longrightarrow \mathbb{F}_{p^k}, \quad \alpha \longmapsto \alpha^p$$

This map preserves addition and multiplication

- $\tau(\alpha + \beta) = \tau(\alpha) + \tau(\beta)$ and $\tau(\alpha \cdot \beta) = \tau(\alpha) \cdot \tau(\beta)$

Let E be an elliptic curve def over $\mathbb{F}_2$ and let $P = (x, y) \in E(\mathbb{F}_{2^k})$

We define a Frobenius map on points in $E(\mathbb{F}_{2^k})$ by

$$\tau(P) = (\tau(x), \tau(y))$$

we are going to claim that τ has the property

$$\tau(P) \in E(\mathbb{F}_{2^k})$$

and if this is true, then so is

$$\tau(P + Q) = \tau(P) + \tau(Q)$$

} τ is a group homomorphism of $E(\mathbb{F}_{2^k})$ to itself

we can prove this by applying the Frobenius map to the generalized Weierstrass equation and factoring the coefficients out to find that

$$\tau(P) = (\tau(x), \tau(y)) \text{ is a point of } E(\mathbb{F}_{2^k})$$

This next theorem shows that the Frobenius map is closely related to the number of points in $E(\mathbb{F}_p)$

Theorem 5.29

Let E be an elliptic curve over $\mathbb{F}_p$ and let

$$t = p + 1 - \#E(\mathbb{F}_p)$$

Note that Hasse's ~~theorem~~ theorem says that $|t| \leq \sqrt{p}$

(a) Let α and β be complex roots of the quadratic polynomial $z^2 - tz + p$

Then $|\alpha| = |\beta| = \sqrt{p}$, and for every $k \geq 1$ we have

$$\#E(\mathbb{F}_{p^k}) = p^k + 1 - \alpha^k - \beta^k$$

(b) Let

$$\tau : E(\mathbb{F}_{p^k}) \longrightarrow E(\mathbb{F}_{p^k}), \quad (x, y) \longmapsto (x^p, y^p)$$

be the Frobenius map. Then for every point $Q \in E(\mathbb{F}_{p^k})$ we have

$$\tau^2(Q) - t \cdot \tau(Q) + p \cdot Q = O$$

where $\tau^2(Q)$ denotes the composition $\tau(\tau(Q))$

Savings in regards to time complexity

- Originally, we ~~used~~ expressed n as a sum of powers of 2 and then used the double and add method to compute nP in order to compute a multiple nP of a point P - this required $\approx \log n$ doublings and $1/2 \log n$ additions (for random values of n)
- the refinement of this using both negative and positive powers of 2 reduces the time to $\approx \log(n)$ doublings and $\frac{1}{3} \log n$ additions.

Koblitz's idea is to replace the doubling map with the Frobenius map → this leads to a large savings, b/c it takes much less time to compute $\tau(P)$ than it does $2P$
- the key to this is the action of the Frobenius map on $E(\mathbb{F}_{2^k})$ satisfies a quadratic equation

Def A Koblitz curve is an elliptic curve def. over $\mathbb{F}_2$ by an equation of the form

$$E_a : Y^2 + XY = X^3 + aX^2 + 1$$

with $a \in \{0,1\}$ The discriminant of E_a is $\Delta = 1$

~~we can take the roots~~

Check that

$$E_0(\mathbb{F}_2) = \{(0,1), (1,0), (1,1), \mathcal{O}\}$$

so $\#E_0(\mathbb{F}_2) = 4$ and

$$t = 2 + 1 - \#E_0(\mathbb{F}_2) = -1$$

To apply theorem 5.29, we use the quadratic formula to find the roots of $Z^2 + Z + 2$, which are

$$\frac{-1 + \sqrt{-7}}{2} \quad \text{and} \quad \frac{-1 - \sqrt{-7}}{2}$$

Then theorem 5.29 gives us

$$\#E_0(\mathbb{F}_{2^k}) = 2^k + 1 - \left(\frac{-1 + \sqrt{-7}}{2}\right)^k - \left(\frac{-1 - \sqrt{-7}}{2}\right)^k$$

with this formula, we can easily compute the number of points in $\#E_0(\mathbb{F}_{2^k})$ even for very large values of k

Thm 5.29 (b) tells us that the Frobenius map τ satisfies the equation $\tau^2 + \tau + 2$ when it acts on points of $E(\mathbb{F}_{2^k})$ i.e,

$$\tau^2(P) + \tau(P) + 2P = \mathcal{O} \quad \text{for all } P \in E(\mathbb{F}_{2^k})$$

We now write an integer n as a sum of powers for τ w/ the assumption that $\tau^2 = -2 - \tau$

Say we have n as

$$n = v_0 + v_1\tau + v_2\tau^2 + \dots + v_l\tau^l \quad \text{w/ } v_i \in \{-1, 0, 1\}$$

Then we can compute nP efficiently by

$$nP = (v_0 + v_1\tau + v_2\tau^2 + \dots + v_l\tau^l)P$$

This takes less time b/c it is far easier to compute $\tau^i(P)$ than it is with 2^iP

Theorem 5.30 defines the above formally

Computing $\#E(\mathbb{F}_{2^k})$ for Koblitz curves is very easy However for curves over $\mathbb{F}_{2^k}$, this is more difficult. The SEA algorithm is reasonably efficient at counting the number of points in $E(\mathbb{F}_q)$ for any fields w/ a large number of elements

Chapter 8.8

genus – "number of holes on the cone" (Maybe?)

Hyperelliptic Curve Cryptography

A hyperelliptic curve of genus g is the set of solutions to an equation of the form

$$C: Y^2 = X^{2g+1} + A_1 X^{2g} + \dots + A_{2g} X + A_{2g+1}$$

with the added req. that the polynomial $F(X) = X^{2g+1} + \dots + A_{2g+1}$ has distinct roots

Just as w/ normal elliptic curves, we have the point $\mathcal{O}$ that lives "at infinity". Thus an elliptic curve is a curve of genus 1

In general there is no addition law for indiv. points on an HEC, but it is possible to define an addition law for collections of points.

Roughly we can take 2 collections of g points, eg.

$$\{P_1, P_2, \dots, P_g\} \text{ and } \{Q_1, Q_2, \dots, Q_g\}$$

and add them to get a new set of points

$$\{R_1, R_2, \dots, R_g\}$$

To describe the addition law, we defined a divisor on C to be a formal sum of points

$$n_1[P_1] + n_2[P_2] + \dots n_r[P_r] \text{ with } P_1, \dots, P_2 \in C \text{ and } n_1, \dots, n_r \in \mathbb{Z}$$

The degree of a divisor is the sum of of its multiplicities

$$\deg(D) = \deg\left[n_1(P_1) + n_2(P_2) + \dots + n_r[P_r]\right] = n_1 + n_2 + \dots + n_r$$

We next define the divisor group of C, denoted by $\mathrm{Div}(C)$ to be the set of divisors on C.

We can add and subtract divisors by adding/subtracting the multiplicities of each point. We also let $\mathrm{Div}_0(C)$ be the set of divisors of deg 0

Two divisors D_1 and D_2 are linearly equivalent if

$$D_1 - D_2 = \text{divisor of a function}$$

We write $\mathrm{Jac}_0(C)$ for the set of divisors of degree 0, with the understanding that linearly equivalent divisors are considered to be identical.

$\mathrm{Jac}_0(C)$ with the addition law obtained by adding the multiplicities of points, is called the Jacobian variety of C. It is the higher genus analogue of elliptic curves and their addition laws

A crucial property of $\mathrm{Jac}_0(C)$ is that it can be described as the set of solutions to a sys. of polynomial eq., and the add. law may also be desc. using polyn.

For ~~convience~~ convenice, let $J = \mathrm{Jac}_0(C)$ and $J(\mathbb{F}_p)$ be points on $\mathrm{Jac}_0(C)$ w/ coord. in $\mathbb{F}_p$

Hyperelliptic ~~Curve~~ Curve Discrete Logarithm problem

Given P and Q in $J(\mathbb{F}_p)$, find an integer m such that $Q = mP$

There is an analogue of Hasse's theorem which says

$$\#J(\mathbb{F}_p) = p^g + O(p^{g-\frac{1}{2}})$$

As g gets large, the computational complexity of the addition law becomes large. So using hyperelliptic curves does not give a significant speed advantage. But using them gives us half as large cipher texts and digital signatures using J rather than E.

Great for Radio frequency identification tags and other constrained devices