11.3 Factorization

In 1987, Hendrik Lenstra published the landmark paper [Len87] that describes and analyzes the Elliptic Curve Method (ECM), which is a powerful algorithm for factoring integers using elliptic curves. Lenstra's method is also described in [ST92, §IV.4], [Dav99, §VIII.5], and [Coh93, §10.3].

Lenstra's algorithm is well-suited for finding "medium sized" factors of an integer N, which today means 10 to 20 decimal digits. The ECM method is not directly useful for factoring RSA challenge numbers (see Section 3.1.3), but surprisingly it is used in intermediate steps of some the algorithms that are used for hunting for such factorizations. Implementation of ECM typically requires little memory. Lenstra's discovery of ECM was inspired by Pollard's $(p-1)$-method, which we will describe in Section 11.3.1 below.

Lenstra

11.3.1 Pollard's $(p-1)$-Method

Definition 11.3.1 (Power-smooth). Let B be a positive integer. A positive integer n is *B-power smooth* if all prime powers dividing n are less than or equal to B.

Thus 30 is 7-power smooth and 5-power smooth, but 4 is not 2-power smooth.

Let N be a positive integer that we wish to factor. We use the Pollard $(p-1)$-method to look for a nontrivial factor of N as follows. First we choose a positive integer B, usually $\leq 10^6$ in practice. Suppose that there is a prime divisor p of N such that $p-1$ is B-power smooth. We try to find p computationally using the following strategy. If $a > 1$ is an integer not divisible by p then by Theorem 3.3.14,

$$a^{p-1} \equiv 1 \pmod{p}.$$

Letting $m = \mathrm{lcm}(1, 2, 3, \ldots, B)$, our assumption that $p-1$ is B-power smooth implies that $p-1 \mid m$, so

$$a^m \equiv 1 \pmod{p}.$$

Thus

$$p \mid \gcd(a^m - 1, N) > 1.$$

If $\gcd(a^m - 1, N) < N$ also then $\gcd(a^m - 1, N)$ is a nontrivial factor of N. If $\gcd(a^m - 1, N) = N$, then $a^m \equiv 1 \pmod{q^r}$ for every prime power divisor q^r of N. In this case, repeat the above steps but with a smaller choice of B or possibly a different choice of a. Also, check from the start whether or not N is not a perfect power M^r, and if so replace N by M.

For fixed B, this algorithm usually splits N when N is divisible by a prime p such that $p-1$ is B-power smooth. Only approximately 15% of primes p in the interval from 10^{15} and $10^{15}+10000$ are such that $p-1$ is 10^6 power-smooth, so the Pollard method with $B=10^6$ already fails nearly 85% of the time at finding 15-digit primes in this range. We will not analyze Pollard's method further, since it was mentioned here only to set the stage for the ECM.

The following examples illustrate the Pollard $(p-1)$-method.

Example 11.3.2. In this example, Pollard works perfectly. Let $N=5917$. We try to use the Pollard $p-1$ method with $B=5$ to split N. We have $m=\mathrm{lcm}(1,2,3,4,5)=60$; taking $a=2$ we have

$$2^{60}-1 \equiv 3416 \pmod{5917}$$

and

$$\gcd(2^{60}-1, 5917) = \gcd(3416, 5917) = 61,$$

so 61 is a factor of 5917.

Example 11.3.3. In this example, we replace B by larger integer. Let $N=779167$. With $B=5$ and $a=2$ we have

$$2^{60}-1 \equiv 710980 \pmod{779167},$$

and $\gcd(2^{60}-1, 779167)=1$. With $B=15$, we have $m=\mathrm{lcm}(1,2,\ldots,15)=360360$,

$$2^{360360}-1 \equiv 584876 \pmod{779167},$$

and

$$\gcd(2^{360360}-1, N) = 2003,$$

so 2003 is a nontrivial factor of 779167.

Example 11.3.4. In this example, we replace B by a smaller integer. Let $N=4331$. Suppose $B=7$, so $m=\mathrm{lcm}(1,2,\ldots,7)=420$,

$$2^{420}-1 \equiv 0 \pmod{4331},$$

and $\gcd(2^{420}-1, 4331)=4331$, so we do not obtain a factor of 4331. If we replace B by 5, Pollard's method works:

$$2^{60}-1 \equiv 1464 \pmod{4331},$$

and $\gcd(2^{60}-1, 4331)=61$, so we split 4331.

Example 11.3.5. In this example, $a=2$ does not work, but $a=3$ does. Let $N=187$. Suppose $B=15$, so $m=\mathrm{lcm}(1,2,\ldots,15)=360360$,

$$2^{360360}-1 \equiv 0 \pmod{187},$$

and $\gcd(2^{360360}-1, 187)=187$, so we do not obtain a factor of 187. If we replace $a=2$ by $a=3$, then Pollard's method works:

$$3^{360360}-1 \equiv 66 \pmod{187},$$

and $\gcd(3^{360360}-1, 187)=11$. Thus $187 = 11\cdot 17$.

11.3.2 Motivation for the Elliptic Curve Method

Fix a positive integer B. If $N = pq$ with p and q prime and $p-1$ and $q-1$ are not B-power smooth, then the Pollard $(p-1)$-method is unlikely to work. For example, let $B = 20$ and suppose that $N = 59 \cdot 101 = 5959$. Note that neither $59 - 1 = 2 \cdot 29$ nor $107 - 1 = 2 \cdot 53$ is B-power smooth. With $m = \mathrm{lcm}(1, 2, 3, \ldots, 20) = 232792560$, we have

$$2^m - 1 \equiv 5944 \pmod{N},$$

and $\gcd(2^m - 1, N) = 1$, so we do not find a factor of N.

As remarked above, the problem is that $p-1$ is not 20-power smooth for either $p = 59$ or $p = 101$. However, notice that $p - 2 = 3 \cdot 19$ is 20-power smooth. Lenstra's ECM replaces $\mathbb{F}_p^\times$, which has order $p-1$, by the group of points on an elliptic curve E over $\mathbb{F}_p$. By Theorem 11.1.4,

$$\#E(\mathbb{F}_p) = p + 1 \pm s$$

for some nonnegative integer $s < 2\sqrt{p}$ and any s can occur. For example, if E is the elliptic curve

$$y^2 = x^3 + x + 54$$

over $\mathbb{F}_{59}$ then by enumerating points one sees that $E(\mathbb{F}_{59})$ is cyclic of order 57 (every abelian group of order 57 is cyclic). The set of numbers $59 + 1 \pm s$ for $s \leq 15$ contains 14 numbers that are B-power smooth for $B \leq 20$. For example, $60 = 59 + 1 + 0$ is 5-power smooth and $70 = 59 + 1 + 10$ is 7-power smooth.

11.3.3 The Elliptic Curve Method

The following description of the ECM algorithm is taken from [Len87], with slight changes to the notation and wording.

> The new method is obtained from Pollard's $(p-1)$-method by replacing the multiplicative group $\mathbb{F}_p^\times$ by the group of points on a random elliptic curve. To find a non-trivial divisor of an integer $N > 1$, one begins by selecting an elliptic curve E over $\mathbb{Z}/N$, a point P on E with coordinates in $\mathbb{Z}/N$, and an integer $m = \mathrm{lcm}(2, 3, \ldots, B)$. Using the addition law of the curve, one next calculates the multiple $m \cdot P$ of P. One now hopes that there is a prime divisor p of N for which $m \cdot P$ and the neutral element $\mathcal{O}$ of the curve become the same modulo p; if E is given by a Weierstrass equation $y^2 = x^3 + ax + b$, with $\mathcal{O} = (0 : 1 : 0)$, then this is equivalent to the third coordinate of $m \cdot P$ being divisible by p. Hence one hopes to find a non-trivial factor of N by calculating the greatest common divisor of this third coordinate with m.

If the above algorithm fails with a specific elliptic curve E, there is an option that is unavailable with Pollard's $(p-1)$-method. We may repeat the above algorithm with a different choice of E. The number of points on E over $\mathbb{Z}/p$ is of the form $p+1-t$ for some t with $|t| < 2\sqrt{p}$, and the algorithm is likely to succeed if $p+1-t$ is B-power-smoth.

Suppose that $P = (x_1, y_1)$ and $Q = (x_2, y_2)$ are nonzero points on an elliptic curve $y^2 = x^3 + ax + b$ and that $P \neq \pm Q$. Let $\lambda = (y_1 - y_2)/(x_1 - x_2)$ and $\nu = y_1 - \lambda x_1$. Recall from Section 10.2.8 the explicit formula for computing $P+P$ and also that $P+Q = (x_3, y_3)$ where

$$x_3 = \lambda^2 - x_1 - x_2 \qquad \text{and} \qquad y_3 = -\lambda x_3 - \nu.$$

We try to compute mP using the powering algorithm from Section 3.5.2. If at some step we can not compute $2^i P + 2^j P$ because we can not compute the inverse modulo N of $x_1 - x_2$, or we can not compute $2^i P$ because we can not compute the inverse of y_1 modulo N, then we compute the gcd of N and $x_1 - x_2$ or y_1. With luck, this gcd is a nontrivial divisor of N.

11.3.4 Examples

For simplicity, we use an elliptic curve of the form

$$y^2 = x^3 + ax + 1,$$

which has the point $P = (0,1)$ already on it.

We factor $N = 5959$ using ECM, then we factor a much larger integer. Let

$$m = \operatorname{lcm}(1, 2, \ldots, 20) = 232792560 = 1101111000000010000111110000_2,$$

where x_2 means x is written in binary. First we choose $a = 1201$ at random and consider $y^2 = x^3 + 1201x + 1$ over $\mathbb{Z}/5959$. Using the formula for $P+P$ from Section 10.2.8 (implemented on a computer) we compute $2^i \cdot P = 2^i \cdot (0,1)$ for $i \in B = \{4,5,6,7,8,13,21,22,23,24,26,27\}$. Then $\sum_{i\in B} 2^i P = mP$. It turns out during no step of this computation does a number not coprime to 5959 appear in any denominator, so we do not split N using $a = 1201$.

The following is MAGMA program that implements ECM.

```
// Returns either 2*P or GCD(N,x1-x2) =/= 1
function double(P,a,N)
   x,y,z := Explode(P);
   if z eq 0 then  // point at infinity
      return P;
   end if;
   g,_,y2inv := XGCD(N,Integers()!(2*y));
   if g ne 1 then
      return g;
   end if;
   xx := ((x^2-a)^2 - 8*x)*y2inv^2;
   yy := ((3*x^2 + a)*(x - xx) - 2*y^2)*y2inv;
   return [xx,yy,1];
end function;
```

```
// Returns P + Q or GCD(N,x1-x2) =/= 1
function add(P,Q,a,N)
   if P eq Q then
      return double(P,a,N);
   end if;
   x1,y1,z1 := Explode(P);
   x2,y2,z2 := Explode(Q);
   if z1 eq 0 then
      return Q;
   elif z2 eq 0 then
      return P;
   end if;
   if x1 eq x2 and y1 eq -y2 then
      return [0,1,0];
   end if;
   g,_,inv := XGCD(N,Integers()!(x1-x2));
   if g ne 1 then
      return g;
   end if;
   lambda := (y1-y2)*inv;
   nu := y1 - lambda*x1;
   x3 := lambda^2 -x1-x2;
   y3 :=  -lambda*x3-nu;
   return [x3,y3,1];
end function;

// Try to compute R=m*[0,1,1] on y^2=x^3+ax+1; returns
// either R or GCD(N,some denominator) =/= 1 if not possible.
function multiply(m,a,N)
   // Points are represented as triples [x,y,z] with z either 0 or 1.
   P := [IntegerRing(N)|0,1,1];
   R := [IntegerRing(N)|0,1,0];
   while m ne 0 do         // computes binary expansion of m.
      if IsOdd(m) then     // if binary digit of m is 1.
         R := add(R,P,a,N);
         if Type(R) eq RngIntElt then
            return R;
         end if;
      end if;
      m := Floor(m/2);
      P := double(P,a,N);
      if Type(P) eq RngIntElt then
         return P;
      end if;
   end while;
   return R;
end function;
```

```
intrinsic ECM1(N::RngIntElt, m::RngIntElt,
               a::RngIntElt) -> RngIntElt
{Try to find a B-power smooth factor of N using Lenstra's ECM
with given a and m=lcm(1,...,B). Returns N on failure.}
   printf "Trying a = %o: \t", a;
   if GCD(4*a^3 + 27, N) ne 1 then
      print "Split using discriminant.";
      return GCD(4*a^3 + 27, N);
   end if;
   R := multiply(m,a,N);
   if Type(R) eq RngIntElt then
      printf "Failed to compute mP. ";
      if R lt N then
         print "Split using denominator.";
         return R;
      end if;
         print "Denominator gives no factor.";
   end if;
   print "Computed mP (no factor found).";
   return N;
end intrinsic;


intrinsic ECM(N::RngIntElt, B::RngIntElt,
       maxtries::RngIntElt) -> RngIntElt
{Try to find a B-power smooth factor of N using Lenstra's ECM.
Returns N on failure. Stop after maxtries tries.}
   m := LCM([1..B]);
   for i in [1..maxtries] do
      a := Random(N);
      M := ECM1(N,m,a);
      if M ne N then
         return M;
      end if;
   end for;
   print "Max tries exceeded.  Trying changing B.";
   return N;
end intrinsic;
```

Exercises

11.1 Let E be an elliptic curve over $\mathbb{F}_p$. Prove that $E(\mathbb{F}_p)$ is a finitely generated abelian group.

11.2 Make up a simple example that illustrates how to use the ElGamal elliptic curve cryptosystem (see Lecture 29). You may mention Nikita and Michael if you wish. Be very clear about what you are illustrating so that the grader can effortlessly understand your example.