# Lenstra's Elliptic Curve Factorization Algorithm

Cryptanalysis

2016.11

## Lenstra's ECM

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

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

- What is it about the quantity $p-1$ that makes it so important for Pollard's method?
- The answer lies in Fermat's little theorem



## Lenstra's ECM

- Intrinsically, $p-1$ is important because there are $p-1$ elements in $\mathbb{F}_p^*$, so every element $\alpha$ of $\mathbb{F}_p^*$ satisfies $\alpha^{p-1} = 1$
- The points and the addition law for an elliptic curve $E(\mathbb{F}_p)$ are very much analogous to the elements and the multiplication law for $\mathbb{F}_p^*$
- Hendrik Lenstra made this analogy precise by devising a factorization algorithm that uses the group law on an elliptic curve



## Lenstra's ECM

- To describe Lenstra's algorithm, we need to work with an elliptic curve modulo $N$, where the integer $N$ is not prime, so the ring $\mathbb{Z}/N\mathbb{Z}$ is not a field
- Start with an equation

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

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

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

# Lenstra's ECM

- Then we can apply the elliptic curve addition formula to compute $2P, 3P, 4P, \ldots$, since the only operations required by that algorithm are addition, subtraction, multiplication, and division (by numbers relatively prime to $N$)



# Lenstra's ECM

- <u>Example</u>. Let $N = 187$ and consider the elliptic curve
$$E: Y^2 = X^3 + 3X + 7$$
modulo 187
- Let $P = (38, 112)$ on $E$ modulo 187
- In order to compute $2P$ mod 187, we follow the elliptic curve addition formula and compute



# Lenstra's ECM

- $\frac{1}{2y(P)} = \frac{1}{224} \equiv 91 \pmod{187}$
- $\lambda = \frac{3x(P)^2 + A}{2y(P)} = \frac{4335}{224} \equiv 34 \cdot 91 \equiv 102 \pmod{187}$
- $x(2P) = \lambda^2 - 2x(P) = 10328 \equiv 43 \pmod{187}$
- $y(2P) = \lambda\big(x(P) - x(2P)\big) - y(P)$
$= 102(38 - 43) - 112 \equiv 126 \pmod{187}$
- Thus $2P = (43, 126)$ as a point on the curve $E$ modulo 187



# Lenstra's ECM

- We can compute $3P = 2P + P$ in a similar fashion and obtain $3P = (54, 105)$
- Also, $4P = (93, 64)$ can be computed by using either $3P + P$ or $2P + 2P$
- Now we attempt to compute $5P = 3P + 2P$ on the elliptic curve

# Lenstra's ECM

- The first step in computing $5P = 3P + 2P$ is to compute the reciprocal of
$$x(3P) - x(2P) = 54 - 43 = 11 \quad \text{modulo } 187.$$
- However, when we apply the extended Euclidean algorithm to 11 and 187, we find that
$$\gcd(11, 187) = 11$$
- So 11 does not have a reciprocal modulo 187



# Lenstra's ECM

- It seems that we have hit a dead end, but in fact, we have struck it rich!
- Notice that since the quantity $\gcd(11, 187)$ is greater than 1, it gives us a divisor of 187
- So our failure to compute $5P$ also tells us that 11 divides 187, which allows us to factor 187 as
$$187 = 11 \cdot 17$$
- This idea underlies Lenstra's elliptic curve factorization algorithm



# Lenstra's ECM

- If we instead look at the elliptic curve $E$ modulo 11, then a quick computation shows that the point
$$P = (38, 112) \equiv (5, 2) \pmod{11}$$
satisfies $5P = \mathcal{O}$ in $E(\mathbb{F}_{11})$
- This means that at some stage of the calculation we have tried to divide by zero
- That is, we are actually trying to find the reciprocal modulo 11 of some integer that is divisible by 11



# Lenstra's ECM

- We replace multiplication modulo $N$ in Pollard's factorization method with addition modulo $N$ on an elliptic curve
- We start with an elliptic curve $E$ and a point $P$ on $E$ modulo $N$ and we compute
$$2! \cdot P,\ 3! \cdot P,\ 4! \cdot P,\ 5! \cdot P, \ldots \pmod{N}.$$
- Notice that once we have computed
$$Q = (n-1)! \cdot P,$$
it is easy to compute $n! \cdot P = nQ$

# Lenstra's ECM

- At each stage, there are three things may happen

1. We are able to compute $n! \cdot P$
2. We need to find the reciprocal of a number $d$ that is a multiple of $N$, which would not be helpful, but luckily this situation is quite unlikely to occur
3. We need to find the reciprocal of a number $d$ that satisfies $1 < \gcd(d, N) < N$, where $\gcd(d, N)$ is a nontrivial factor of $N$, so we are happy



# Lenstra's ECM

- A minor problem is to find an initial point $P$ on an elliptic curve $E$ modulo $N$
- The obvious method is to fix an equation for the curve $E$, plug in values of $X$, and check whether the quantity $X^3 + AX + B$ is a square modulo $N$
- Unfortunately, this is difficult to do unless we know how to factor $N$



# Lenstra's ECM

- The solution to this dilemma is to first choose the point $P = (a, b)$ at random
- Second, choose a random value for $A$
- Third, set

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

- Then the point $P$ is automatically on the curve $E: Y^2 = X^3 + AX + B$ modulo $N$



# Lenstra's ECM

**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 elliptic curve $E : Y^2 = X^3 + AX + B$.

**3.** Loop $j = 2, 3, 4, \ldots$ up to a specified bound.

- **4.** Compute $Q \equiv jP \pmod{N}$ and set $P = Q$.
- **5.** If computation in Step **4** fails, then we have found a $d > 1$ with $d \mid N$.
- **6.** If $d < N$, then **success**, return $d$.
- **7.** If $d = N$, go to Step **1** and choose a new curve and point.

**8.** Increment $j$ and loop again at Step **2**.

# Lenstra's ECM

- Example. We illustrate Lenstra's algorithm by factoring $N = 6887$
- We begin by randomly selecting a point
  $$P = (1512, 3166)$$
  and a number $A = 14$ and computing
  $$B \equiv 3166^2 - 1512^3 - 14 \cdot 1512 \equiv 19 \pmod{6887}.$$
- We let $E$ be the elliptic curve
  $$E: Y^2 = X^3 + 14X + 19$$
  and $P$ is automatically on $E$ modulo 6887



# Lenstra's ECM

- Now we start computing multiples of $P$ modulo 6887
- First we find that
  $$2P \equiv (3466, 2996) \pmod{6887}$$
- Next we compute
  $$3!\,P = 3(2P) \equiv (3067, 396) \pmod{6887}$$
- And so on



# Lenstra's ECM

| $n$ | $n! \cdot P \bmod 6887$ | | |
|---|---|---|---|
| 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 | $6! \cdot P$ | $=$ | $(6141, 5581)$ |



# Lenstra's ECM

- It is only when we try, and fail, to compute $7! \cdot P$, that something interesting happens
- Let $Q = 6! \cdot P = (6141, 5581)$, and we want to compute $7Q$
- First we compute
  $$2Q \equiv (5380, 174) \pmod{6887},$$
  $$4Q \equiv 2 \cdot 2Q \equiv (203, 2038) \pmod{6887}.$$

# Lenstra's ECM

- Then we compute $7Q$ as

$$\begin{aligned} Q &\equiv (Q + 2Q) + 4Q \\ &\equiv \big((6141, 5581) + (5380, 174)\big) + (203, 2038) \\ &\equiv (984, 589) + (203, 2038) \qquad \pmod{6887} \end{aligned}$$

- When we attempt to perform the final step, we need to compute the reciprocal of $203 - 984$ modulo 6887
- But we find that $\gcd(203 - 984, 6887) = 71$



# Lenstra's ECM

- Thus we have discovered a nontrivial divisor of 6887, namely 71, which gives the factorization

$$6887 = 71 \cdot 97$$

- In $E(\mathbb{F}_{71})$, the point $P$ satisfies $63P \equiv \mathcal{O}$, while in $E(\mathbb{F}_{97})$, the point $P$ satisfies $107P \equiv \mathcal{O}$
- The reason that we succeeded in factoring 6887 using $7! \cdot P$ is precisely because $7!$ is the smallest factorial that is divisible by 63



# Lenstra's ECM

- Remark. It is an interesting and useful property of the elliptic curve factorization algorithm that its expected running time depends on the smallest prime factor of $N$, rather than on $N$ itself
- More precisely, if $p$ is the smallest factor of $N$, then the elliptic curve factorization algorithm has average running time approximately

$$O\left(e^{\sqrt{2(\log p)(\log\log p)}}\right) \quad \text{steps}$$



# Lenstra's ECM

- If $N = pq$ is a product of two primes with $p \approx q$, the running times of ECM and QS are approximately equal, and then the fact that a sieve step is much faster than an elliptic curve step makes sieve methods faster in practice
- However, the elliptic curve method is quite useful for finding moderately large factors of extremely large numbers, because its running time depends on the smallest prime factor

# Factoring of $2^{1039} - 1$

## 2 Selecting a kilobit SNFS target number

Once the decision had been reached to attempt a kilobit SNFS factorization by a joint effort, it remained to find a suitable target number to factor. In this section we describe the process that led to our choice of $2^{1039} - 1$.

Regular RSA moduli were ruled out, since in general they will not have the special form required for SNFS. Special form numbers, however, are not especially concocted to have two factors of approximately the same size, and have factors of a priori unknown sizes. In particular, they may have factors that could relatively easily be found using factoring methods different from SNFS, such as Pollard's $p - 1$ or $\rho$ method, or the elliptic curve method (ECM, cf. [11]). Thus, for all kilobit special form numbers under consideration, we first spent a considerable ECM effort to increase our confidence that the number we would eventually settle for would not turn out to have an undesirably small factor, i.e., a factor that could have been found easier using, for instance, ECM.

http://eprint.iacr.org/2007/205