CERIAS

FFT-ECM by Division Polynomials for Factoring

Zhihong Li & Samuel S. Wagstaff, Jr.

Introduction

Factoring large numbers is very useful in cryptography. One can break certain ciphers, such as the RSA cryptosystem, if one can factor large numbers. In this poster we develop a new integer factoring algorithm similar to the ECM(elliptic curve method). The difference is that this algorithm uses division polynomials and a FFT(fast Fourier transform) to compute multiples of many points simultaneously. The ordinary ECM has little chance of factoring an RSA public key and breaking the cipher. This algorithm has a much greater chance of factoring a number of the size of RSA public keys currently used.

Review of Elliptic Curves and the ordinary ECM:

An elliptic curve is the graph $E_{a,b}$ of an equation $y^2 = x^3 + ax + b$, where *x,y,a,b* are real numbers, rational numbers or integers modulo m>1. The set $E_{a,b}$ also contains a point at infinity, denoted ∞. The point is not a point on the graph of $yE_{a,b}x^3 + ax + b$. It is the identity of the elliptic curve group. If *P=(x,y)* lies on the graph of $y^2 = x^3 + ax + b$, we define *–P=(x,-y)*. Given two points *P* and *Q*, on the graph but not on the same vertical line, define *P+Q=-R*, where *R* is the third point on the straight line through *P* and *Q*. If the tangent line through *P* is vertical, then we define $P + P = \infty$. We can prove that an elliptic curve, with group operation defined above, is a group.

In 1985, H.W.Lenstra,Jr. invented an ingenious new factoring algorithm that uses elliptic curves. It performs a calculation *mP*, where *m* is the product of all primes less than some bound *B* raised to some suitable power. One then computes *mP* and hopes that it will equal the identity of $E_{a,b}$ modulo *p*, which is a prime factor of the integer we need to factor, but will not equal the identity of $E_{a,b}$ modulo *n*, which is the integer we need to factor.

The FFT-ECM by Division Polynomials:

We create a new, much faster ECM using division polynomials and fast Fourier transforms. First of all, let's define division polynomials.

Def: (of Division Polynomials)

Define *division polynomials* inductively as follows:

$$\begin{aligned}
&\psi_1 = 1, \psi_2 = 2y,\\
&\psi_3 = 3x^4 + 6ax^2 + 12bx - a^2,\\
&\psi_4 = 4y(x^6 + 5ax^4 + 20bx^3 - 5a^2x^2 - 4abx - 8b^2 - a^3),\\
&\psi_{2m+1} = \psi_{m+2}\psi_m^3 - \psi_{m-1}\psi_{m+1}^3 \qquad (m \ge 2),\\
&2y\psi_{2m} = \psi_m(\psi_{m+2}\psi_{m-1}^2 - \psi_{m-2}\psi_{m+1}^2) \qquad (m \ge 3).
\end{aligned}$$

Further define

$$\begin{aligned}
&\phi_m = x\psi_m^2 - \psi_{m+1}\psi_{m-1},\\
&4y\omega_m = \psi_{m+2}\psi_{m-1}^2 - \psi_{m-2}\psi_{m+1}^2.
\end{aligned}$$

Prop:

Suppose *E* is an elliptic curve given by $y^2 = x^3 + ax + b$ with $\Delta = -16(4a^3 + 27b^2) \neq 0$ and , then the multiple of *P* is given by

$$[m]P = \left(\frac{\phi_m(P)}{\psi_m(P)^2}, \frac{\omega_m(P)}{\psi_m(P)^3}\right).$$

The algorithm relies on the following observations:

1. Pick a random pair of integers *x, y,* then for any chosen *a,* let $b = y^2 - x^3 - ax$. This yields univariate polynomials $\psi_m(a)$ in the variable *a.* I proved that the degree of $\psi_m(a)$

$$\deg(\overline{\psi}_m) = \begin{cases} \frac{m^2}{4} - 1 & \text{if } m \text{ is even} \\ \frac{m^2 - 1}{4} & \text{if } m \text{ is odd} \end{cases}.$$

2. The ECM finds the prime factor *p* of *n* when computing *[B]P* for $a = 2^i \bmod n, i = 1, \ldots, B^2$, if and only if $\gcd(\overline{\psi}_B(2^i), n) = p$ for some .

3. One may evaluate a polynomial at all terms of a geometric progression by using discrete Fourier transforms to compute the convolution of two polynomials, and this may be done swiftly by FFT. In fact, such an FFT takes only arithmetic steps without division.

Sketch of the Algorithm:

To find a prime factor *p* of *n,* set *B* (I'm working on finding suitable *B* by testing the algorithm). Pick $x, y \in Z_n$, compute $\overline{\psi}_B(2), i = 1, \ldots, B^2$ and check that $\gcd(\overline{\psi}_B(2^i), n)$ is non-trivial. If the *gcd* is non-trivial, then we are done. If the *gcd* is n then partition the product into subproducts and compute individual *gcd*'s. If the *gcd* is 1, then we should pick up another *x,y* and *B*. In this algorithm, we try curves in a time.

The Algorithm for Evaluation of Polynomial on Geometric Progression:

We want to evaluation the polynomial $x(t) = \sum_{j=0}^{D-1} x_j t^j$ at values $t_k = T^k, 0 \le k \le D-1$, *T* is a constant (in the algorithm here, *T*=2). We transform the sums according to

$$\begin{aligned}
\sum_j x_j T^{kj} &= T^{-k^2/2} \sum_j (x_j T^{-j^2/2}) T^{(-k-j)^2/2}\\
&= T^{\Delta_n} \sum_j (x_j T^{\Delta_j}) T^{-\Delta_{k+j}},
\end{aligned}$$

Where $\Delta_n = n(n+1)/2$. By this transform, we can evaluate the polynomial by calculate the convolution of two vectors.

Here is the algorithm.

Input: An integer *T* and coefficients $x_0, \ldots, x_{d-1}$ of a polynomial *x(t)*. (We assume *T* has an inverse in the arithmetic domain.)

Output: The values for $x(T^k), k = 0, \ldots, d-1$.

Choose $C = 2^c$ with least *c* such that $C \ge 2d$.

For (*j=0* to *d-1*) {

$x_j = x_j T^{\Delta_j}$

}

Zero-pad $x = (x_j)$ to have length *C*.

Let .

Perform the length-C cyclic convolution $z = x \otimes y$ by FFTs.

Return $(x(T^k)) = (T^{\Delta_{k-1}} z_{C/2+k-1}), k \in [0, D-1]$.

The convolution in the penultimate line of the algorithm may be computed with discrete Fourier Transforms as

$$x \otimes y = DFT^{-1}(DFT(x) * DFT(y))$$

Where * is pointwise multiplication of vectors. The *DFT*'s are performed efficiently by *FFT*'s.

Significance of the method:

This new factoring method we are developing will be able to find much larger prime factors of large Integers than the ordinary ECM can. The ordinary ECM often finds 40-digit factors and sometimes finds 50-digit factors of large integers. Its record discovery is a 57-digit prime factor. A heuristic argument similar to that used to estimate the complexity of the ordinary ECM suggests that our new method will easily find 60-digit factors, will often find 75-digit factors, and occasionally find 90-digit factors.

Several important ciphers, such as those of RSA and of Rabin and Williams, could be broken if one could factor large integers quickly. Each discovery of a new integer factoring algorithm either increases the size of the keys that must be used in these ciphers or imposes some other restriction on them. In the case of the new factoring method proposed here, the restriction is that the composite key must not have a prime factor in the range that could be found by this method.

Our new factoring method would also be an enormous help to those who research the mathematical problems that require the explicit factorization into primes of many large numbers.

PURDUE UNIVERSITY

CERIAS®

Discovery Park e-Enterprise Center