Integer Factoring: Algorithms and Implementation

C.E.Veni Madhavan

Dept. of Computer Science & Automation
Indian Institute of Science
Bangalore-560 012, India
cevm@csa.iisc.ernet.in

B-IT, University of Bonn, Germany
23 May 2006

There are essentially two fundamental principles on which all the methods of factoring an integer N are based. They are

1. Find nontrivial solutions to the equation $x^2 = y^2 + N$ or to the congruence $x^2 \equiv y^2 \pmod{N}$ and check $\gcd(x \pm y, N)$.

2. Compute the powers a^k of a randomly chosen element a in an appropriate group and check the $\gcd(a^k - 1, N)$. ∎

Currently all general purpose large-scale integer factoring efforts are based on the mathematical, algorithmic ideas and the implementation nuances behind the number field sieve technique. Our current work is based on these. In my talk I will present the details of some of our recent work in this area.

The major steps in the NFS comprise;

1. *Polynomial Selection:* The choice of the number field is crucial for the asymptotic speed-up. This is carried out by choosing a *suitable* polynomial f for constructing the number field. The choice is dependent on various characteristics, namely (i) extension degree (usually 5 for factoring integers of sizes less than 768 bits), (ii) the coefficients of f, usually required to be *skewed*, i.e.

the coefficients increase in size, gradually from the leading coefficient c_5 to the trailing coefficient c_0, (iii) *required* densities of number of roots of f modulo the prime integers of the factor base and (iv) relationships with the densities of *large primes* arising in the sieving stage. All these characteristics are modeled based on analytic and empirical considerations and matched with experimental results. We are developing various techniques to handle these issues and to come out with best possible outcomes.

2. *Sieving:* This is the major compute intensive stage. Computations in the scale of 2^{70} cycles are required. Many exciting possibilities spanning a combination of hardware accelerators, PC and super computer networks are deployed to great effect. Various implementation considerations based on time-memory trade-offs, number of processors, handling *very large* size factor bases determine the practical efficiencies. We have devised special algorithms and data structures to handle these issues. Further practical challenges are concerning the *management* of immense computations in a fail-safe, fault-tolerant, robust cluster/grid computing environments. We are developing methodologies to handle this.

3. *Linear Algebra:* This stage consists of solving immensely large system of sparse linear equations over $\mathbf{F}_2$. A combination of Gaussian elimination and block Lanczos iterative methods are used with delicate balancing of sizes, sparsity and packing techniques. We are enhancing known techniques with *newer* heuristics.

4. *Square roots of algebraic integers:* This is an interesting exercise in computing with very large number of polynomials in *low degree* polynomial rings with *large* coefficients. Custom designed efficient techniques are being developed to handle this.

We have obtained several refinements and enhancements of the NFS algorithm, concerning the above steps. We are employing these ideas to carry out the factoring of the RSA-640, RSA-704 challenge numbers. I shall detail our current work on these in the talk.