# On the Complexity of Computing Discrete Logarithms and Factoring Integers

*A. M. Odlyzko*

Bell Laboratories
Murray Hill, New Jersey 07974

Practically all knapsack public key cryptosystems have been broken in the last few years, and so essentially the only public key cryptosystems that still have some credibility and are widely known are those whose security depends on the difficulty of factoring integers (the RSA scheme and its variants) and those whose security depends on the difficulty of computing discrete logarithms in finite fields. Therefore, the computational complexity of these two problems is of great interest.

At the time of the workshop, one aspect of the then-current state of knowledge on these two fundamental problems seemed to be highly unsatisfactory. This was the fact that all of the fast algorithms for discrete logarithms and all but one of the fast algorithims for factoring integers had running time estimates that depended on the efficiency with which matrices could be inverted. These algorithms require the solution of a system of linear equations of the form

$$Ax = y \ , \tag{1}$$

where $A$ is a matrix of size $m$ by $n$, $x$ and $y$ are column vectors of lengths $m$ and $n$, respectively, and $m$ is close to $n$. The interesting ranges of values for $n$ are between $10^3$ and $10^7$. Ordinary gaussian elimination requires that about $n^3$ steps for the solution of (1). Strassen's algorithm, which might be practical for large $n$, takes about $n^{\log_2 7} = n^{2.807...}$ steps. The best general purpose algorithm that is known, due to Coppersmith and Winograd [3], takes about $n^{2.495...}$ steps, but is almost certainly impractical. No algorithm can solve the system (1) in fewer than about $n^2$ steps (there are that many entries in the matrix, after all!).

Depending on how fast the system (1) can be solved, various algorithms have different asymptotic running time estimates. If we let $L = L(p)$ denote any quantity that satisfies

$$L = \exp((1 + o(1)) \ (\log_e p \ \ \log_e \log_e p))^{1/2}) \quad \text{as} \quad p \to \infty \ , \tag{2}$$

and suppose that the system (1) can be solved in time about $n^r$ for various values of $r$, then Table 1 summarizes the state of knowledge at the time of the workshop about the efficiency of the best factoring algorithms for factoring integers around $p$ in size. A similar table can be prepared for the running times of various discrete logarithm algorithms.

Table 1. Asymptotic Running Times for Factoring Integers.

| algorithm | $r = 3$ | $r = 2.807...$ | $r = 2.495...$ | $r = 2$ |
|---|---|---|---|---|
| Schnorr-Lenstra [9] | $L$ | $L$ | $L$ | $L$ |
| Continued fraction [8] | $L^{1.13...}$ | $L^{1.12...}$ | $L^{1.11...}$ | $L^{1.11...}$ |
| Schroeppel linear sieve [8] | $L^{1.22...}$ | $L^{1.18...}$ | $L^{1.11...}$ | $L$ |
| Pomerance quadratic sieve [8] | $L^{1.06...}$ | $L^{1.04...}$ | $L^{1.02...}$ | $L$ |
| Coppersmith, Odlyzko, and Schroeppel [2] | $L^{1.16...}$ | $L^{1.13...}$ | $L^{1.081...}$ | $L$ |

The question that was raised at the workshop was whether the estimates for the running times of those algorithms that are obtained by assuming $r > 2$ are really appropriate. Even if we cannot solve general systems of the form (1) in time $O(n^{2+\varepsilon})$ for every $\varepsilon > 0$, we can take advantage of the fact that the systems that arise in factorization and discrete logarithm algorithms are very sparse. Some methods to take advantage of that sparseness were presented, and their effectiveness was supported both by large-scale simulation and heuristic arguments. (See [7] for a brief description.) The conclusion was drawn that at least in the foreseeable future, these methods are likely to make the system (1) easy to solve. Still, a question remained about the asymptotic performance.

As a result of that presentation, several methods were developed that can solve sparse systems of the form (1) in not much more than $n^2$ steps. The first such methods were developed by D. Coppersmith and the author, following a suggestion of N. Karmarkar. These methods consist of adaptations of the conjugate gradient [4] and the Lanczos [5] algorithms to solve linear equations over finite fields. They have been tested successfully on quite large systems. A brief account of these adaptations is given in [7].

Soon afterwards, D. Wiedemann [10] found a more elegant and probably even faster method, based on the use of Berlekamp-Massey algorithm and the Cayley-Hamilton theorem. A brief account of it can also be found in [7].

Now that the main question, whether systems of the form (1) that arise in factorization and discrete logarithm algorithms can be solved in time about $n^2$, has been answered in the affirmative, we are faced with a more important and basic question. There are now several algorithms known that can factor an integer around $p$ in time $L(p)$ (see Table 1 and [6], which presents a new algorithm based on elliptic curves), as well as several algorithms that can compute discrete logarithms in fields $GF(p)$ for $p$ a prime in time $L(p)$. (For fields $GF(2^n)$, discrete logarithms can be computed much faster [1].) Does this mean that $L(p)$ is the natural lower bound for the computational complexity of factoring and finding discrete logarithms? It is this author's guess that this is not the case, and that we are missing some insight that will let us break below the $L(p)$ barrier.

## REFERENCES

[1] D. Coppersmith, Fast evaluation of logarithms in fields of characteristic two, *IEEE Trans. Inform. Theory* **IT-30** (1984), 587-594.

[2] D. Coppersmith, A. M. Odlyzko, and R. Schroeppel Discrete logarithms in $GF(p)$, *Algorithmica*, to appear.

[3] D. Coppersmith and S. Winograd, On the asymptotic complexity of matrix multiplication, *SIAM J. Comp.* **11** (1982), 472-492.

[4] M. R. Hestenes and E. Stiefel, Methods of conjugate gradients for solving linear systems, *J. Res. Nat. Bureau of Standards* **49** (1952), 409-436.

[5] C. Lanczos, Solution of systems of linear equations by minimized iterations, *J. Res. Nat. Bureau of Standards* **49** (1952), 33-53.

[6] H. W. Lenstra, Jr., manuscript in preparation.

[7] A. M. Odlyzko, Discrete logarithms in finite fields and their cryptographic significance, *Proc. EUROCRYPT 84*, to appear.

[8] C. Pomerance, Analysis and comparison of some integer factoring algorithms, pp. 89-139 in *Computational Number Theory: Part 1,* H. W. Lenstra, Jr., and R. Tijdeman, eds., Math. Centre Tract 154, Math. Centre, Amsterdam, 1982.

[9] C. P. Schnorr and H. W. Lenstra, Jr., A Monte Carlo factoring algorithm with linear storage, *Math. Comp.* **43** (1984), 289-311.

[10] D. Wiedemann, Solving sparse linear equations over finite fields, *IEEE Trans. Inform. Theory*, to appear.