Computation of Discrete Logarithms in Prime Fields

(Extended Abstract†)

*B. A. LaMacchia**
A. M. Odlyzko

AT&T Bell Laboratories
Murray Hill, New Jersey 07974

1. Introduction

If p is a prime and g and x integers, then computation of y such that

$$y \equiv g^x \bmod p, \quad 0 \leq y \leq p-1 \tag{1.1}$$

is referred to as *discrete exponentiation*. Using the successive squaring method, it is very fast (polynomial in the number of bits of $|p| + |g| + |x|$). On the other hand, the inverse problem, namely, given p, g, and y, to compute some x such that Equation 1.1 holds, which is referred to as the *discrete logarithm* problem, appears to be quite hard in general. Many of the most widely used public key cryptosystems are based on the assumption that discrete logarithms are indeed hard to compute, at least for carefully chosen primes.

The current state of knowledge about discrete logarithms is surveyed in [3, 4]. If certain precautions concerning the choice of p are observed, then the best published algorithms [1] for computing discrete logarithms modulo a prime p have running time

$$\exp((1 + o(1))(\log p)^{\frac{1}{2}}(\log\log p)^{\frac{1}{2}}) \quad \text{as } p \to \infty. \tag{1.2}$$

The estimate given by Equation 1.2 is of roughly the same form as that of most of the fast practical algorithms for factoring composite integers of about the same size as p. An important feature of the estimate above is that it applies to a precomputation phase that has to be carried out once for each prime p. Once that phase is completed, individual discrete logarithms modulo that prime are much easier to compute.

---

†Full text of this paper to appear in *Designs, Codes, and Cryptography* **1** (1991).
*Present address: MIT, Cambridge, MA 02139

A.J. Menezes and S.A. Vanstone (Eds.): Advances in Cryptology - CRYPTO '90, LNCS 537, pp. 616-618, 1991.

3. Sieving and Linear Algebra

We used a sieve to find pairs of integers (c_1, c_2) whose residues $c_1V - c_2T$ were smooth over the small real primes in Q. For each pair (c_1, c_2) with a smooth residue, the corresponding equation $c_1V - c_2T \equiv V(c_1 + c_2s) \bmod p'$ was factored (if possible) over the extended (complex) factor base. The sieve considered approximately 2.7×10^{10} (c_1, c_2) pairs, of which 288,017 yielded equations.

The various algorithms that are available for solving large sparse linear systems over finite fields are surveyed in [2]. Here we will only mention briefly how they performed on our problem. The system of 288,017 equations in 96,321 unknowns was reduced by the structured Gaussian elimination method to a smaller system of 7,262 equations in 6,006 unknowns. The resulting smaller system was then solved modulo 2 using a conjugate gradient program, and modulo $\frac{p-1}{2}$ using the Lanczos algorithm. These results suggest that linear algebra is likely to be a significant but not an insurmountable problem in computing discrete logarithms modulo large primes.

4. Conclusions

Our experiments with the 192-bit Sun prime as well as with a 224-bit prime demonstrate that the discrete log algorithms of [1] are indeed practical. Using the same amount of computing power that is used to factor a generally hard 115 decimal digit integer by the multiple polynomial quadratic sieve method, one can compute discrete logarithms modulo a prime of at least 100 decimal digits.

References

[1] D. Coppersmith, A. Odlyzko, and R. Schroeppel, Discrete logarithms in $GF(p)$, *Algorithmica* **1** (1986), 1-15.

[2] B. A. LaMacchia and A. M. Odlyzko, Solving large sparse linear systems over finite fields, *Advances in Cryptology: Proceedings of Crypto '90*, A. Menezes, S. Vanstone, eds., to be published.

[3] K. S. McCurley, The discrete logarithm problem, in *Cryptography and Computational Number Theory*, C. Pomerance, ed., *Proc. Symp. Appl. Math.*, Amer. Math. Soc., 1990, to appear.

[4] A. M. Odlyzko, Discrete logarithms in finite fields and their cryptographic significance, *Advances in Cryptology: Proceedings of Eurocrypt '84*, T. Beth, N. Cot, I. Ingemarsson, eds., *Lecture Notes in Computer Science* **209**, Springer-Verlag, NY (1985), 224-314.