A THEOREM ON FACTORING POLYNOMIALS OVER FINITE FIELDS

by

Victor Shoup

Computer Sciences Technical Report #866

August 1989

A Theorem on Factoring Polynomials over Finite Fields

Victor Shoup
Computer Sciences Department
University of Wisconsin
Madison, WI 53706

August 4, 1989

Let p be a prime number, and let $S(p-1)$ denote the largest prime divisor of $p-1$. In this note, we prove the following theorem.

Theorem 1 *Assuming the Extended Riemann Hypothesis (ERH), we can deterministically factor a polynomial of degree n over $\mathbf{F}_p$ in time $S(p-1)^{1/2}(n \log p)^{O(1)}$.*

The algorithm we describe is a refinement of algorithms given by von zur Gathen [9] and Rónyai [7]. Assuming the ERH, these algorithms run in time $S(p-1)(n \log p)^{O(1)}$, thus our algorithm represents an improvement of a factor of $S(p-1)^{1/2}$. If the ERH is true, then in terms of the dependence on p, the bound on the running time of our algorithm is better than the worst-case bounds on the running times of current algorithms in the literature.

The algorithms of von zur Gathen and Ronyai essentially reduce the problem of factoring a polynomial of degree n over $\mathbf{F}_p$ to the following three problems in time $(n \log p)^{O(1)}$:

(1) computing the prime factorization of $p-1$;

(2) computing q-th roots in $\mathbf{F}_p$ for primes $q \mid p-1$;

(3) computing the roots of polynomials $g \in \mathbf{F}_p[X]$, where

(a) $\deg g \leq n$,

(b) g is a divisor of $X^q - a$ ($a \in \mathbf{F}_p, q$ a prime divisor of $p-1$), and

(c) we are given a q-th root of a.

Both of these algorithms solve problem (2) using a variant of the root finding algorithm of Adleman, Manders and Miller [1]. Using well-known discrete logarithm techniques, this algorithm can be implemented so as to run in time $q^{1/2}(\log p)^{O(1)}$. These algorithms solve problem (3) by simple brute-force search through all of the q roots of $X^q - a$, requiring time $q(n \log p)^{O(1)}$ in the worst case.

In the algorithm that we describe here, we eliminate the need to solve problems of type (2) (at least for primes $q > n$), and we employ a variant of the technique used in the Pollard-Strassen integer factoring algorithm [6, 8] to solve problem (3) in time $q^{1/2}(n \log p)^{O(1)}$.

We now describe our algorithm.

Let

$$p - 1 = q_1^{e_1} \cdots q_r^{e_r}$$

be the prime factorization of $p-1$. This factorization can certainly be obtained deterministically in within the stated time bound by means of the Pollard-Strassen factoring algorithm.

Let $f \in \mathbf{F}_p[X]$ be the polynomial we wish to factor. It will suffice to demonstrate how to find a nontrivial divisor of f. By making use of well-known deterministic polynomial time reductions, we can assume that

$$f = (X - a_1)(X - a_2) \cdots (X - a_n),$$

where the a_i's are disinct elements of $\mathbf{F}_p$ [4, 5].

Let $R = \mathbf{F}_p[X]/(f)$. Let $x = X \bmod f$ be the image of x in R. By the Chinese Remainder Theorem, the map that takes $y \in R$ to

$$(y \bmod (X - a_1), \ldots, y \bmod (X - a_n)) \in \bigoplus_{i=1}^{n} \mathbf{F}_p$$

is an $\mathbf{F}_p$-algebra isomorphism. In the discussion that follows, we shall simply identify R and $\bigoplus_i \mathbf{F}_p$ without explicitly mentioning this isomorphism. Under this correspondence, we have $x = (a_1, \ldots, a_n)$. Elements in $\mathbf{F}_p \subset R$ are of the form $(a, \ldots, a)$ (all components equal). To find a nontrivial factor of f, it

will suffice to find an element $u \in R$ of the form $(u_1, \ldots, u_n)$ where some, but not all, of the u_i's are zero; viewing u now as a polynomial over $\mathbf{F}_p$, $\gcd(f, u)$ is a nontrivial divisor of f. Let's call such an element u a "splitter."

Let a be an element in $\mathbf{F}_p^*$. Then by group theory, a can be expressed uniquely as

$$a = a^{(1)}a^{(2)} \cdots a^{(r)},$$

where $a^{(j)}$ is an element in $\mathbf{F}_p^*$ of order dividing $q_j^{e_j}$. In fact, given the factorization of $p-1$, this representation of a is efficiently computable by the formula $a^{(j)} = a^{(p-1)/q_j^{e_j}}$.

Now, for $j = 1, \ldots, r$, $x^{(p-1)/q_j^{e_j}} = (a_1^{(j)}, \ldots, a_n^{(j)})$. Since, the a_i's are all different, there must be some $j = 1, \ldots, r$ such that $y = x^{(p-1)/q_j^{e_j}} \notin \mathbf{F}_p$. We can easily compute such a j along with the corresponding y.

Let's fix $q = q_j, e = e_j$. We have $y = (y_1, \ldots, y_n)$, where the y_i's are elements in $\mathbf{F}_p^*$ of order dividing q^e, not all of which are the same. Since $y^{q^e} = 1$, we can easily compute the least t $(1 \leq t \leq e)$ such that $a = y^{q^t} \in \mathbf{F}_p$. Let $z = y^{q^{t-1}}$, which we surely computed as a biproduct in computing a. Note that $z = (z_1, \ldots, z_n)$, where the z_i's are q-th roots of a, not all the same. It will suffice to discover just one of the z_i's, since $z - z_i$ is a splitter.

To find a component of z, we reduce the problem of factoring f to yet another factoring problem. Using linear algebra, in time $(n \log p)^{O(1)}$ we can compute the least degree monic polynomial $g \in \mathbf{F}_p[X]$ such that $g(z) = 0$. One can easily prove that g is of the form

$$g = (X - z_1') \cdots (X - z_m'),$$

where $\{z_1', \ldots, z_m'\}$ is the set of distinct elements among $z_1, \ldots, z_n$. So we have now reduced our problem to finding a root of g.

To solve this problem, we first find a q-th root of a in $\mathbf{F}_p$. If $q \mid m$, then we can use the algorithm of Adleman, Manders and Miller to find a q-th root of a in time $(n \log p)^{O(1)}$. Otherwise, we can find a q-th root of a in time $(n \log p)^{O(1)}$ as follows. Suppose that the constant term of g is b, and the multiplicative inverse of m mod q^e is $\bar{m}$ (which we can compute in time $(\log p)^{O(1)}$). Then we claim that $((-1)^m b)^{\bar{m}}$ is a q-th root of a. To see this, note that we can write $g = (X - \xi_1\alpha) \cdots (X - \xi_m\alpha)$, where α is a q-th root of a and the ξ_i's are q-th roots of unity. Therefore, the constant term of g is $(-1)^m \xi' \alpha^m$, where ξ' is a q-th root of unity. Since α has order dividing q^e, we have $((-1)^m b)^{\bar{m}} = (\xi')^{\bar{m}}\alpha$, which is another q-th root of a.

So we have reduced our problem to finding a root of g, where g divides $X^q - a$, and where we already know one q-th root of a, call it α. We can solve this problem in time $q^{1/2}(n \log p)^{O(1)}$ using the following procedure, which is a variation of the Pollard-Strassen integer factoring technique.

We shall require a primitive q-th root of unity. The work performed so far may have already yielded such an element, but if not, under the assumption of the ERH, with Ankeny's theorem we can obtain in time $(\log p)^{O(1)}$ a single primitive q-th root of unity, call it ξ [3]. Let $S = \mathbf{F}_p[X]/(g)$, and let $\lambda = X \bmod g \in S$ be the image of X in S. Let $s = \lfloor q^{1/2} \rfloor$. Consider the polynomials

$$h_i(X) = (X - \xi^{si}\alpha)(X - \xi^{si+1}\alpha) \cdots (X - \xi^{si+(s-1)}\alpha) \qquad (i = 0, \ldots s-1).$$

If we could compute all of the h_i's, then we could examine them one at a time until we found one for which $\gcd(g, h_i) \neq 1$. If we succeeded in finding such an h_i, then we could search for a root of g in the set $\{\xi^{si}\alpha, \xi^{si+1}\alpha, \ldots, \xi^{si+(s-1)}\alpha\}$ (which has s elements); otherwise, we could search in the set $\{\xi^{s^2}\alpha, \xi^{s^2+1}\alpha, \ldots, \xi^{q-1}\alpha\}$ (which has $\leq 2q^{1/2} - 1$ elements).

It will suffice to compute the h_i's mod g. To do this, we first compute the polynomial $h(X) = (X-1)(X-\xi)\cdots(X-\xi^{s-1}) \in \mathbf{F}_p[X]$. Using the FFT, this takes time $s(\log s)^{O(1)}$ [2]. But note that

$$\begin{aligned} h_i(\lambda) &= (\lambda - \xi^{si}\alpha) \cdots (\lambda - \xi^{si+(s-1)}\alpha) \\ &= (\xi^{si}\alpha)^s (\lambda/\xi^{si}\alpha - 1) \cdots (\lambda/\xi^{si}\alpha - \xi^{s-1}) \\ &= (\xi^{si}\alpha)^s h(\lambda/\xi^{si}\alpha). \end{aligned}$$

So to compute the h_i's mod g, it suffices to evaluate the polynomial $h(X)$ at s points in S, which can be done using the FFT with $s(\log s)^{O(1)}$ additions, subtractions, and multiplications in S [2], each of which can be performed in time $(n \log p)^{O(1)}$.

References

[1] L. Adleman, K. Manders, and G. Miller. On taking roots in finite fields. In *18th Annual Symposium on Foundations of Computer Science*, pages 175–178, 1977.

[2] A. V. Aho, J. E. Hopcroft, and J. D. Ullman. *The Design and Analysis of Computer Algorithms.* Addison-Wesley, Reading, 1974.

[3] N. C. Ankeny. The least quadratic nonresidue. *Ann. of Math.*, 55:65–72, 1952.

[4] E. R. Berlekamp. Factoring polynomials over large finite fields. *Math. Comp.*, 24(111):713–735, 1970.

[5] R. Lidl and H. Niederreiter. *Finite Fields.* Addison-Wesley, Reading, 1983.

[6] J. Pollard. Theorems on factorization and primality testing. *Proc. Cambridge Phil. Soc.*, 76:521–528, 1974.

[7] L. Rónyai. Factoring polynomials modulo special primes. To appear, *Combinatorica*, 1988.

[8] V. Strassen. Einige Resultate über Berechnungskomplexität. *Jahresber. Deutsch. Math.-Verein*, 78:1–8, 1976.

[9] J. von zur Gathen. Factoring polynomials and primitive elements for special primes. *Theoret. Comput. Sci.*, 52:77–89, 1987.