Smoothness and Factoring Polynomials over Finite Fields*

Victor Shoup†
Computer Sciences Department
University of Toronto
Toronto, Ontario M5S 1A4, Canada

February 15, 1996

Keywords: polynomials, finite fields, factorization.

Let p be a prime number, and $\mathbf{F}_p$ the finite field with p elements. Let $S(m)$ be the "smoothness" function that for integers m is defined as the largest prime divisor of m. In this note, we prove the following theorem.

Theorem. *There is a deterministic algorithm for factoring polynomials over* $\mathbf{F}_p$*, which on polynomials over* $\mathbf{F}_p$ *of degree* n *runs in time*

$$S(p-1)^{1/2}(n \log p)^{O(1)}$$

under the assumption of the Extended Riemann Hypothesis (ERH).

The algorithm we describe is a refinement of algorithms given by von zur Gathen [19] and Rónyai [13]. 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 for factoring arbitrary polynomials over $\mathbf{F}_p$. See [15] for an unconditional running time bound of $p^{1/2}(n \log p)^{O(1)}$, and [8, 14] for running time bounds (assuming ERH) of $(n \log p)^{O(1)}$ for polynomials of a special form.

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 two problems in time $(n \log p)^{O(1)}$:

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

(2) computing *all* of the roots of $X^q - a$, where q is a prime divisor of $p-1$, and a is a q-th power residue in $\mathbf{F}_p$.

*Appeared in *Information Processing Letters* 39, pp. 39–42, 1991.

†This research was done while the author was at the Unviversity of Wisconsin–Madison with the support of NSF grants DCR-8504485 and DCR-8552596. An earlier version of this paper appeared under the title "A Theorem on Factoring Polynomials over Finite Fields" as Computer Sciences Technical Report No. 866, University of Wisconsin–Madison, August 1989.

Both of these algorithms solve problem (2) using a variant of the root finding algorithm of Adleman, Manders and Miller [1]. Using discrete logarithm techniques in [11], a *single* q-th root of a can be computed in time $q^{1/2}(\log p)^{O(1)}$ assuming the ERH. But these algorithms in general compute all q of the q-th roots of a, which of course requires time at least q. Actually, only a very small subset of these roots is required, and our algorithm computes this subset in time $q^{1/2}(n \log p)^{O(1)}$ by employing a variant of the technique used in the Pollard-Strassen integer factoring algorithm [12, 18]. We also note that problem (1) can be solved in time $S(p-1)^{1/2}(\log p)^{O(1)}$ using the Pollard-Strassen integer factoring algorithm.

Remark 1. The ERH assumption in our theorem can be replaced by the assumption that we have a primitive root mod p. The following *unconditional* consequence of this was pointed out by I. E. Shparlinskiy [17]. It was proved in [9] that for a positive proportion of primes p, $S(p-1) \leq p^{\alpha+\epsilon}$, where $\alpha = 1/(2\sqrt{e}) = 0.303\ldots$. Combining this fact with the bound in [6] on the average order of the least primitive root, it follows that for a positive proportion of primes p, we can factor any polynomial of degree n over $\mathbf{F}_p$ in deterministic time $p^{\alpha/2+\epsilon}n^{O(1)} = p^{0.155}n^{O(1)}$.

Remark 2. One can prove an analog of our theorem with $S(p+1)$ replacing $S(p-1)$ by making use of the methods in this paper and in [4], along with [16, Theorem 2].

Preliminaries

Suppose $f \in \mathbf{F}_p[X]$ is the polynomial we wish to factor. Using well-known techniques [5, 10], we can assume that f is of the form

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

where the a_i's are distinct elements of $\mathbf{F}_p$. Furthermore, we need only address the problem of finding a nontrivial factor of f, since an algorithm for this problem can obviously be used to completely factor f.

Let R be the $\mathbf{F}_p$-algebra $\mathbf{F}_p[X]/(f)$. Let x denote the residue class of X modulo f in R. Then any element in R can be written as $g(x)$ where g is a polynomial of degree less than n over $\mathbf{F}_p$, and by the Chinese Remainder Theorem, the map that sends $g(x)$ to

$$(g(a_1), \ldots, g(a_n)) \in \underbrace{\mathbf{F}_p \oplus \cdots \oplus \mathbf{F}_p}_{n \text{ times}}$$

is an $\mathbf{F}_p$-algebra isomorphism. We shall identify elements $u \in R$ with their image under this isomorphism, writing $u = (u_1, \ldots, u_n)$, with each u_i in $\mathbf{F}_p$, and referring to the u_i's as the *components* of u. In particular, $x = (a_1, \ldots, a_n)$, and for elements a in $\mathbf{F}_p$—which is as a subalgebra of R under the usual embedding sending $a \in \mathbf{F}_p$ to $a \bmod f$—we have $a = (a, \ldots, a)$ (all components equal to a). This isomorphism will only be used in the analysis of our algorithm, and not in the algorithm itself—indeed, computing this isomorphism is equivalent to factoring f. The algorithm itself represents elements of R by polynomials in $\mathbf{F}_p[X]$ of degree less than n.

The zero divisors of R are precisely those $u = (u_1, \ldots, u_n) \in R$ where some, but not all, of the u_i's are zero. If g is a polynomial over $\mathbf{F}_p$ with $g(x) = u$, then $\gcd(f, g)$ is a nontrivial divisor of f—in fact,

$$\gcd(f, g) = \prod_{\substack{1 \leq i \leq n \\ u_i = 0}} (X - a_i).$$

Thus, the problem of finding a nontrivial divisor of f reduces to the problem of finding a zero divisor in R.

For a given $u \in R$, there exists a unique monic polynomial of minimum degree in $\mathbf{F}_p[X]$ that is zero at u, and we denote this polynomial by M_u. The degree of M_u is no more than n, and we can easily compute M_u using linear algebra by looking for the first linear relation among the powers $1, u, u^2, \ldots$. Furthermore, it is easy to see that if $u = (u_1, \ldots, u_n)$, then

$$M_u = \prod_{a \in \{u_i\}} (X - a),$$

where the product ranges over all of the *distinct* components of u.

Let

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

be the prime factorization of $p-1$. Since $\mathbf{F}_p^*$ is a cyclic group, from elementary group theory we know that $\mathbf{F}_p^*$ can be written as the inner direct of subgroups $S_1, \ldots, S_r$, where S_j is of order $q_j^{e_j}$. Therefore, any $a \in \mathbf{F}_p^*$ can be written uniquely as

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

where $a^{(j)} \in S_j$. Given the prime factorization of $p-1$, we can easily compute $a^{(j)}$ using the formula $a^{(j)} = a^{b_j}$, where $b_j = c_j d_j$, $d_j = (p-1)/q_j^{e_j}$, and c_j is the multiplicative inverse of d_j modulo $q_j^{e_j}$.

Factoring Algorithm

We now describe our algorithm. The notation described in the Preliminaries section remains in force. As described there, it suffices to find a zero divisor in the ring R associated with f.

Step 1. *Factor $p-1$ and compute b_j, $j = 1, \ldots, r$.*

The factorization of $p-1$ can be obtained in time $S(p-1)^{1/2}(\log p)^{O(1)}$ using the Pollard-Strassen integer factoring algorithm.

Step 2. *Compute x^{b_j}, $j = 1, \ldots, r$, and from among these select one x^{b_j} that is not in $\mathbf{F}_p$. For this j, let $y = x^{b_j}$, $q = q_j$, $e = e_j$.*

For any given j, $x^{b_j} = (a_1^{(j)}, \ldots, a_n^{(j)})$. Since the a_i's are all different, there must be some $j = 1, \ldots, r$ such that the components of x^{b_j} are not all the same, i.e. $x^{b_j} \notin \mathbf{F}_p$.

Step 3. *Compute the least t such that $y^{q^t} \in \mathbf{F}_p$. Let $a = y^{q^t}$, and $z = y^{q^{t-1}}$.*

Since $y \notin \mathbf{F}_p$ and $y^{q^e} = 1$, we know that t as defined above satisfies $1 < t \leq e$. Observe that $z = (z_1, \ldots, z_n)$, where the z_i's are q-th roots of a in $\mathbf{F}_p$, not all the same (since t was chosen so that $z \notin \mathbf{F}_p$).

The purpose of the remaining steps of the algorithm is to find one of the z_i's, since for any z_i, $z - z_i$ is a zero divisor. If it happens that $q \leq n$, then we can compute all of the roots of $X^q - a$ in time $(n \log p)^{O(1)}$ (assuming the ERH) using the algorithm of Adleman, Manders and Miller, and then easily find a zero divisor. Therefore, in the sequel we shall assume that $q > n$.

Step 4. *Compute M_z, the minimum polymial of z. Let $m = \deg M_z$.*

We know that

$$M_z = (X - z'_1) \cdots (X - z'_m),$$

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 M_z. Before we do this, however, we do the following.

Step 5. *Compute a q-th root α of a in $\mathbf{F}_p$.*

We can find a q-th root of a in time $(n \log p)^{O(1)}$ as follows (without assuming the ERH). Suppose that the constant term of M_z is b, and that the multiplicative inverse of m mod q^e is $\bar{m}$ (which exists since we are assuming that $q > n \geq m$, and which we can compute in time $(\log p)^{O(1)}$). Then we claim that $\alpha = ((-1)^m b)^{\bar{m}}$ is a q-th root of a. To see this, note that we can write $M_z = (X - \xi_1 \alpha') \cdots (X - \xi_m \alpha')$, where α' is some q-th root of a and the ξ_i's are q-th roots of unity. Therefore, $b = (-1)^m \xi' (\alpha')^m$, where ξ' is some 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 also a q-th root of a.

Step 6. *Compute a root of M_z.*

We can find a root of M_z in time $q^{1/2}(n \log p)^{O(1)}$ as follows. We shall require a primitive q-th root of unity, call it ξ. Under the assumption of the ERH, with Ankeny's theorem [3] we can obtain ξ in time $(\log p)^{O(1)}$.

If q were small, i.e., $q = (n \log p)^{O(1)}$, we could simply search among the elements $\xi^i \alpha$, $i = 0, \ldots, q-1$, for a root of M_z. However, for large q, we can use the following variant of the Pollard-Strassen integer factoring technique.

Let $S = \mathbf{F}_p[X]/(M_z)$, and let λ denote the residue class of X modulo M_z 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 $\gcd(M_z, h_i) \neq 1$. If we succeeded in finding such an h_i, then we could search for a root of M_z 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 M_z. 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 $s(\log p)^{O(1)}$ (see, e.g., [2, 7]). 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 M_z, 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 (again, see [2, 7]), each of which can be performed in time $(n \log p)^{O(1)}$.

References

[1] L. M. Adleman, K. Manders, and G. L. 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, 1974.

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

[4] E. Bach and J. von zur Gathen. Deterministic factorization of polynomials over special finite fields. Technical Report 799, Computer Sciences Department, University of Wisconsin–Madison, 1988.

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

[6] D. A. Burgess and P. D. T. A. Elliot. The average of the least primitive root. *Mathematika*, 15:39–50, 1968.

[7] D. G. Cantor and E. Kaltofen. Fast multiplication of polynomials over arbitrary rings. Technical Report 87-35, Department of Computer Science, Rensselaer Polytechnic Institute, 1987. To appear, *Acta. Inf.*

[8] S. A. Evdokimov. Factoring a solvable polynomial over a finite field and Generalized Riemann Hypothesis. *Zapiski Nauchn. Semin. Leningr. Otdel. Matem. Inst. Acad. Sci. USSR*, 176:104–117, 1989. In Russian.

[9] J. B. Friedlander. Shifted primes without large prime factors. In *Number Theory and Applications*, pages 393–401. Kluwer Acad. Publ., 1989.

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

[11] S. Pohlig and M. Hellman. An improved algorithm for computing logarithms over GF(p) and its cryptographic significance. *IEEE Trans. Inf. Theory*, 24:106–110, 1978.

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

[13] L. Rónyai. Factoring polynomials modulo special primes. *Combinatorica*, 9(2):199–206, 1989.

[14] L. Rónyai. Galois groups and factoring polynomials over finite fields. In *30th Annual Symposium on Foundations of Computer Science*, pages 99–104, 1989.

[15] V. Shoup. On the deterministic complexity of factoring polynomials over finite fields. *Inform. Process. Lett.*, 33(5):261–267, 1990.

[16] V. Shoup. Searching for primitive roots in finite fields. In *22nd Annual ACM Symposium on Theory of Computing*, pages 546–554, 1990. To appear, *Math. Comp.*

[17] I. E. Shparlinskiy. Personal Communication, 1990.

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

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