MATHEMATICS OF COMPUTATION
VOLUME 36, NUMBER 154
APRIL 1981

A New Algorithm for Factoring Polynomials Over Finite Fields*

By David G. Cantor and Hans Zassenhaus

Abstract. We present a new probabilistic algorithm for factoring polynomials over finite fields.

1. Introduction. Suppose F is a finite field of characteristic p with $q = p^d$ elements. A fundamental computational task is to find the irreducible factors of a monic polynomial

$$f(t) = \sum_{i=0}^{n} f_i t^i, \quad f_n = 1, \tag{1}$$

in $F[t]$. There are two standard methods; one is due to E. Berlekamp, the other appears to be a "folk method". Both are described by D. Knuth in [3, pp. 381–397]. Further improvements, for special values of q, are given by R. Moenck in [5]. See also E. Berlekamp [1]. (Note that both methods as described in the references apply only when $d = 1$, however, straightforward modifications, described below, allow $d > 1$.) Both methods use the calculation of resultants (or equivalently the solution of linear equations) to reduce the problem to finding the roots of a polynomial which has all of its roots in F. When p is very small, probabilistic methods are used. An improvement to the "folk method" method, along with a more explicit calculation of the work required, has recently been given by M. Rabin [6].

We present here a new probabilistic method which, when combined with the above algorithms, avoids the need for both resultants and linear equations. It leads to algorithms which are conceptually simpler than previous methods. Moreover, it works equally well for all finite fields F, regardless of the magnitude of q. When used for factoring a quadratic $x^2 - a$, it reduces to Berlekamp's algorithm. Other standard algorithms for factoring quadratics are due to D. H. Lehmer [4] and D. Shanks [7]. Our algorithm is also suitable for finding solutions of polynomial equations over finite fields.

2. Preliminaries. The first part of our method is a slight variant of the first part of the "folk method" mentioned above. The variant allows $d > 1$. Initially, we remove the multiple factors from f. If the formal derivative $f'(t)$ is 0, then $f(t)$ has the form

$$f(t) = \sum_{i=0}^{n/p} f_i t^{ip} = g(t)^p,$$

Received February 4, 1980; revised July 11, 1980.

1980 *Mathematics Subject Classification.* Primary 12C05, 12E05, 68C05.

Key words and phrases. Factoring, polynomials, finite fields, probabilistic algorithms.

*The preparation of this paper was sponsored in part by NSF Grant MCS 79-03711.

with

$$g(t) = \sum_{i=0}^{n/p} f_i^{q/p} t^i,$$

and it is enough to factor $g(t)$. If $f'(t)$ is not zero, then we may compute $h(t) = \gcd(f(t), f'(t))$, and replace $f(t)$ by $f(t)/h(t)$. We repeatedly perform the above two operations until $\gcd(f(t), f'(t)) = 1$. In this way, we obtain a polynomial with the same irreducible factors as the original polynomial, but with each factor occurring only once. Next, we reduce the factorization problem to the case when all irreducible factors of f have the same degree. To do this, we define $f_1(t) = f(t)$ and inductively for $j = 1, 2, 3, \ldots$ define $u_j(t) = \gcd(f_j(t), t^{q^j} - t)$ and $f_{j+1}(t) = f_j(t)/u_j(t)$. The iteration stops when $f_{j+1}(t)$ is constant. It is easy to see (and well known) that $u_j(t)$ is the product of all the irreducible factors, of degree j, of $f(t)$.

In [1], E. Berlekamp concluded that the above reductions may run more slowly than other methods based on matrix reduction. In [2], J. Calmet and R. Loos give computer timing for this type of phenomenon.

3. Separating Factors of Equal Degree: The New Method. In the first part of this section, we assume that $f(t)$ has degree n and is a product of r distinct irreducible factors $u_i(t)$, $1 \leqslant i \leqslant r$, with $\deg(u_i) = s_i$, $\sum_{i=1}^r s_i = n$. Consider the ring $R = F[t]/(f(t))$. Since the $u_i(t)$ are pairwise relatively prime, there exist polynomials $e_i(t)$, $1 \leqslant i \leqslant r$, of degree $< n$, satisfying

$$e_i(t) \equiv \begin{cases} 1 & (\operatorname{mod} u_i(t)), \\ 0 & (\operatorname{mod} u_j(t)), j \neq i. \end{cases} \tag{2}$$

It is clear that

$$e_i(t)e_j(t) \equiv \begin{cases} e_i(t) & (\operatorname{mod} f(t)) \text{ if } i = j, \\ 0 & (\operatorname{mod} f(t)) \text{ if } i \neq j. \end{cases} \tag{3}$$

Furthermore, $\sum_{i=1}^r e_i(t) \equiv 1 \pmod{f(t)}$, and, since each $e_i(t)$ has degree $< n$, we see that $\sum_{i=1}^r e_i(t) = 1$. We thus have the direct sum composition

$$R = \bigoplus Re_i(t) \tag{4}$$

(here and throughout, we shall denote an element of R by a polynomial in its coset; thus $e_i(t)$ denotes the coset $e_i(t) + f(t)F[t]$). The summand $Re_i(t)$, in (2), is a field of q^{s_i} elements and there is a natural isomorphism of the field $F[t]/(u_i(t))$ onto $Re_i(t)$ which sends the polynomial $g(t)$ to $g(t)e_i(t)$ (more precisely it sends the residue class $g(t) + u_i(t)F[t]$ to the residue class $g(t)e_i(t) + f(t)F[t]$, but, as stated above, we shall slur over this technicality in what follows).

The decomposition (4) is, of course, the classical Wedderburn decomposition of the semisimple, commutative ring R.

For the remainder of this section, we assume all s_i are equal, say, to s. Assume first that q is odd and suppose we have found

$$a(t) = \sum_{i=1}^{r} a_i e_i(t) \in R, \tag{5}$$

where each $a_i = 0, \mp 1$, and further suppose that $a(t) \not\equiv 0, \mp 1 \pmod{f(t)}$. Put

$S = \{i\colon a_i = 0\}$ and $T = \{i\colon a_i = 1\}$. Then, if T is not empty, we see that

$$\gcd(f(t), a(t) - 1) = \prod_{i \in T} u_i(t)$$

is a proper factor of $f(t)$, while, if T is empty,

$$\gcd(f(t), a(t)) = \prod_{i \in S} u_i(t)$$

is a proper factor of $f(t)$.

Thus, such $a(t)$ lead to a factorization of $f(t)$. To find such $a(t)$, we proceed at random. Specifically, we choose a random polynomial $b(t)$ from the $q^n - q$ nonconstant polynomials of degree $< n$ in $F[t]$, giving each such polynomial the probability $1/(q^n - q)$. We can write

$$b(t) \equiv \sum_{i=1}^{r} b_i(t)e_i(t) \pmod{f(t)}.$$

Put $m = (q^s - 1)/2$. Then

$$b(t)^m \equiv \sum_{i=1}^{r} b_i(t)^m e_i(t) \pmod{f(t)}.$$

The calculation of each $b_i(t)^m$ may be performed $(\operatorname{mod} u_i(t))$ and thus $b_i(t)^m \equiv 0$, 1, or $-1 \pmod{u_i(t)}$. Unless $b(t)^m \equiv 0, \mp 1 \pmod{f(t)}$, we may put $a(t) \equiv b(t)^m$ $(\operatorname{mod} f(t))$ and obtain a factorization of $f(t)$.

We now calculate the probability that $b(t)^m \equiv 0, \mp 1 \pmod{f(t)}$. Since we chose $b(t)$ nonconstant, a fortiori nonzero, $b(t)^m \not\equiv 0 \pmod{f(t)}$. For each i, there exist m polynomials $b_i(t)$, of degree $< s$, such that $b_i(t)^s \equiv 1 \pmod{u_i(t)}$ and m such that $b_i(t)^s \equiv -1 \pmod{u_i(t)}$. Hence, there exist $2m^r$ polynomials $b(t)$ of degree $< n$ satisfying $b(t) \equiv \mp 1 \pmod{f(t)}$, and $q - 1$ of these will be constant. Therefore, the probability of randomly choosing $b(t)$ satisfying $b(t)^m \equiv \mp 1 \pmod{f(t)}$ is

$$\frac{2m^r - q + 1}{q^n - q} < 2^{1-r} \leqslant 1/2.$$

So, the probability of success at each trial is $> 1 - 2^{1-r} \geqslant 1/2$.

We now consider the case $p = 2$. If $q \equiv 1$ (3), then F contains a primitive 3rd root of unity ρ satisfying $\rho^2 + \rho + 1 = 0$ (and we assume ρ is known). Put $m = (q^s - 1)/3$. Then, choosing $b(t)$ as before, define

$$a(t) \equiv b(t)^m \equiv \sum_{i=1}^{r} a_i e_i(t) \pmod{f(t)},$$

where the $a_i \in \mathrm{GF}(4) = \{0, 1, \rho, \rho^2\}$. Thus, if $a(t) \notin \mathrm{GF}(4) \pmod{f(t)}$, then at least one of $\gcd(f(t), a(t) + 1)$, $\gcd(f(t), a(t) + \rho)$, $\gcd(f(t), a(t))$ will be a nontrivial factor of $f(t)$. We now calculate the probability that $a(t) \in \mathrm{GF}(4)$. Suppose $b(t) \equiv \sum_{i=1}^{r} b_i e_i(t) \pmod{f(t)}$. For $j = 0, 1, 2$, there exist m choices for each b_i satisfying $b_i^m = \rho^j \pmod{u_i(t)}$. Hence, the probability that $a(t) \in \mathrm{GF}(4)$ is

$$\frac{3m^r - q + 1}{q^n - q} < 3^{1-r} \leqslant 1/3.$$

If $q \equiv 2$ (3), we simply factor $f(t)$ in the quadratic extension field $F(\rho)$ of F and then combine factors which are conjugate over F.

4. Applications to Berlekamp's Methods. We now show how our probabilistic method can be applied to Berlekamp's algorithm, replacing the calculation of resultants and allowing explicit computation of the probability of success. In Berlekamp's algorithm, it is assumed that $f(x) = \Pi_{i=1}^{r} f_i(x)$ where the $f_i(x)$ are distinct irreducible polynomials of (not necessarily equal) degree s_i. Thus, only the first, not the second, reduction at the beginning of Section 2 need be applied. Next a basis $v_i(x)$, $1 \leqslant i \leqslant r$, of the r-dimensional F-vector space V, consisting of the solutions $a(x) \pmod{f(x)}$ to the congruence $a(x)^q \equiv a(x) \pmod{f(x)}$, is obtained. We can write any $a(x) \in V$ in the form

$$a(x) = \sum_{i=1}^{r} a_i(x)e_i(x), \tag{6}$$

where the $e_i(x)$ are the polynomials described at the beginning of Section 3 satisfying (2), (3), (4). Then, $a(x)^q \equiv \Sigma_{i=1}^{r} a_i(x)^q e_i(x)$, hence, $a_i(x)^q \equiv a_i(x)$ $(\operatorname{mod} f_i(x))$, and so $a_i = a_i(x) \in F$. We see that V consists of the q^r polynomials of the form $\Sigma_{i=1}^{r} a_i e_i(x)$, where the $a_i \in F$. We now choose for $a(x)$ a random, nonzero element of V. This is done by choosing a nonzero r-tuple $(b_1, b_2, \ldots, b_r)$ $\in F^r$ from the $q^r - 1$, such r-tuples giving each probability $1/(q^r - 1)$, and defining $a(x) = \Sigma_{i=1}^{r} b_i v_i(x)$. Conversely, given $a(x) \in V$, we can write $a(x) = \Sigma_{i=1}^{r} a_i e_i(x)$, and all $q^r - 1$ r-tuples $(a_1, a_2, \ldots, a_r)$ are equally likely.

First consider the case q odd. We form $a(x)^m = \Sigma_{i=1}^{r} a_i^m e_i^m$, where $m = (q - 1)/2$. The a_i^m are 0 or ∓ 1. Unless all a_i^m are equal, either $\gcd(f(x), a(x)^m - 1)$ or $\gcd(f(x), a(x)^m + 1)$ will be a nontrivial factor of $f(x)$. The number of nonzero r-tuples $(a_1, a_2, \ldots, a_r)$ with all a_i^m equal is $2m^r$. Thus, the probability of a nontrivial factorization is at least

$$1 - 2m^r/(q^r - 1) \geqslant 1 - 2^{1-r} \geqslant 1/2.$$

After producing such a factorization, $f(x) = g_1(x)g_2(x)$, one may proceed in two ways.

The first is to apply the above method to g_1 and g_2, separately, and proceed recursively.

The second is to suppose inductively that we have found $t < r$ factors $g_1(x), g_2(x), \ldots, g_t(x)$ such that $\Pi_{i=1}^{t} g_i(x) = f(x)$. Then we choose a new random $a(x)$, as before, and form $\gcd(g_i(x), a(x)^m \mp 1)$, $1 \leqslant i \leqslant t$. This is done most easily by first forming $\gcd(f(x), a(x) \mp 1)$ and then taking the gcd of these two polynomials with each of the $g_i(x)$, one-by-one. This will fail to yield a nontrivial factorization of at least one $g_i(x)$ if and only if, in the expression $a(x)^m = \Sigma_{i=1}^{r} a_j^m e_j(x)$, all the a_j^m, corresponding to those $e_j(x)$ which divide a given $g_i(x)$, are equal. If $g_i(x)$ has r_i factors (where $\Sigma_{i=1}^{r} r_i = r$), then the probability of a failure is less than

$$\left(\prod_{i=1}^{t} (1 + 2m^{r_i})\right) / (q^r - 1) < (1/2)^{r-t},$$

and so the probability that at least one nontrivial factorization occurs is $> 1/2$.

When q is even, one first considers, as in Section 3, the case when $F \supset \mathrm{GF}(4) = \{0, \rho, \rho^2, 1\}$, where ρ is a primitive cube root of 1. Suppose that $f(x)$ has already been factored into $t < r$ factors $f(x) = \Pi_{i=1}^{t} g_i(x)$. In this case, we compute

$\gcd(g_i(x), b(x)^m - \rho^j)$, $m = (q-1)/3$, $1 \leqslant i \leqslant t$, $j = 0, 1, 2$. Denote by r_i the number of irreducible factors of g_i. The probability that no nontrivial factorization of at least one $g_i(x)$ occurs is less than

$$\left(\prod_{i=1}^{t}(1+3m^{r_i})\right)/(q^r-1) < (1/3)^{r-t}.$$

Finally, if $\rho \notin F$, one carries out the factorization in the quadratic extension field $F(\rho)$ and at the end combines conjugate factors.

5. Some Modifications. For fields where $q-1$ has small factors, various modifications of the above procedure are possible. Suppose $h > 1$ is a divisor of $q-1$ and $(h, q) = 1$. Then there exists a primitive hth root of unity ξ in F. Such a ξ can be found explicitly. (For example, factor $x^h - 1$ using the first of the above procedures, since $x^h - 1$ has distinct linear factors. Alternatively, choose a random nonzero element θ of F and compute θ^m, $m = (q-1)/h$. With probability $\phi(h)/h$, θ^m will be a primitive hth root of 1.) Now suppose, as in Section 4, that $f(x)$ has r factors and we have already obtained the factorization $f(x) = \prod_{i=1}^{t} g_i(x)$ where each $g_i(x)$ has r_i factors with $\sum_{i=1}^{t} r_i = r$. We choose $a(x)$, a random nonzero element, of V and form the polynomials $\gcd(g_i(x), a(x)^m - \xi^j)$, $1 \leqslant i \leqslant t$, $0 \leqslant j \leqslant h-1$. The probability that no nontrivial factorization of at least one $g_i(x)$ occurs is

$$\left(-1+\prod_{i=1}^{t}(1+hm^{r_i})\right)/(q^r-1) < \left(\prod_{i=1}^{t}(1+hm^{r_i})\right)/q^r$$
$$= \left(\prod_{i=1}^{t}(1+hm^{r_i})\right)/q^r = \prod_{i=1}^{t}\left((h^{r_i-1}+(q-1)^{r_i})/(q^{r_i}h^{r_i-1})\right)$$
$$< \prod_{i=1}^{t}\left(q^{r_i}/(q^{r_i}h^{r_i-1})\right) = h^{t-r},$$

since $q^{r_i} - (q-1)^{r_i} > r_i(q-1)^{r_i} > h^{r_i-1}$. Thus, the probability of a nontrivial factorization has been increased, at the cost of computing a greater number of gcd's.

Note that if h is a small divisor of $q^d - 1$, we can apply similar modifications to the methods of Section 3.

When $q = p^d$, $d > 1$, there is a useful modification of Berlekamp's method. Suppose first that q is odd. Consider the solutions $a(x) \pmod{f(x)}$ to the congruence $a(x)^p \equiv a(x) \pmod{f(x)}$. Writing such a solution in the form (6), we see that each $a_i(x)$ satisfies $a_i(x)^p \equiv a_i(x) \bmod f_i(x)$, hence $a_i = a_i(x) \in \mathrm{GF}(p)$, the finite field of p elements contained in F. Thus, the solutions $a(x)$ form a vector space U over $\mathrm{GF}(p)$, and a basis for the $a(x)$ may be found by computing $g_i(x) \equiv x^{ip} \bmod(f(x))$ for $0 \leqslant i < n$. The coefficients of $g_i(x)$ may be expressed as linear combinations of elements of a fixed basis of F over $\mathrm{GF}(p)$. This will lead to a system of nd equations in nd unknowns over $\mathrm{GF}(p)$. The probabilistic method may then be applied to random elements $a(x)$ of U, which should be raised to the $(p-1)/2$ power. If $q = 2^d$ is even, we suppose as before that $F \supset \mathrm{GF}(4)$ and, equivalently, that d is even. The above method may then be applied with $\mathrm{GF}(4)$ replacing $\mathrm{GF}(p)$, and in this case the a_i will be $0, 1, \rho, \rho^2$ and exponentiation will not be necessary.

6. Remarks. We first note that the effective cost of algorithms of this nature is extremely hard to estimate, for there are numerous methods of implementation, and the best implementation depends upon the magnitude of the degree n of f, and upon q. The asymptotically best algorithms frequently turn out to be worst on all problems for which they are used. For the problem at hand, the largest amount of work is probably that of calculating high powers of polynomials; e.g. $b(x)^{q^m}$ (mod $f(x)$) may have to be calculated, perhaps, for several $b(x)$. If x^{iq} (mod $f(x)$), $0 \leqslant i < n$, is tabulated, then $b(x)^{q^{j+1}}$ may be obtained from $b(x)^{q^j}$ by one matrix-by-vector multiplication. This yields, using classical techniques, an algorithm whose expected running time is $O(n^3 + n^2 \log q)$. Modern, asymptotically good algorithms may be better asymptotically, but not useful in this context.

Note that most of the procedures described by R. Moenck [5], for special choices of q, apply also to this algorithm.

It is interesting to compare our algorithm with Berlekamp's in the case when $f(x)$ has only linear factors. In this case, Berlekamp's method calls for calculating $\gcd(f(x), (x - c)^{(q-1)/2})$ for random $c \in F$. Our method differs only in that we compute $\gcd(f(x), b(x)^{(q-1)/2})$ for random, nonconstant $b(x) \in F[x]$. Indeed, as noted earlier, when $f(x)$ is a quadratic polynomial, the methods are (essentially) identical. However, the apparently minor change, which allows a general $b(x)$ instead of $(x - c)$, is the key to our new method. For (1) it allows precise probability calculations to be made; (2) it generalizes to the separation of higher-degree factors.

Department of Mathematics
University of California
Los Angeles, California 90024

Department of Mathematics
Ohio State University
Columbus, Ohio 43210

1. E. R. Berlekamp, "Factoring polynomials over large finite fields," *Math. Comp.*, v. 24, 1970, pp. 713–735.

2. J. Calmet & R. Loos, *An Improvement of Rabin's Probabilistic Algorithm for Generating Irreducible Polynomials Over GF(p)*, Interner Bericht Nr. 3/80, Universität Karlsruhe, West Germany.

3. D. E. Knuth, *The Art of Computer Programming, Seminumerical Algorithms*, Vol. 2, Addison-Wesley, Reading, Mass., 1969.

4. D. H. Lehmer, "Computer technology applied to the theory of *n* numbers," *Studies in Number Theory* (W. J. LeVeque, Ed.,), Math. Assoc. of America, 1969.

5. R. T. Moenck, "On the efficiency of algorithms for polynomial factoring," *Math. Comp.*, v. 31, 1977, pp. 235–250.

6. M. O. Rabin, "Probabilistic algorithms in finite fields," *SIAM J. Comput.*, v. 9, 1980, pp. 273–280.

7. D. Shanks, *Five Number Theoretical Algorithms* (Proc. Second Manitoba Conf. on Numerical Math.), University of Manitoba, Winnipeg, Manitoba, Canada, 1972.