J. Symbolic Computation (2001) **31**, 3–17
doi:10.1006/jsco.1999.1002
Available online at http://www.idealibrary.com on IDEAL®

Factoring Polynomials Over Finite Fields: A Survey

JOACHIM VON ZUR GATHEN[†§] AND DANIEL PANARIO[‡¶]

[†] *Fachbereich Mathematik-Informatik, Universität Paderborn, D-33095 Paderborn, Germany*
[‡] *Department of Computer Science, University of Toronto, Toronto, Ontario, Canada M5S 3G4*

This survey reviews several algorithms for the factorization of univariate polynomials over finite fields. We emphasize the main ideas of the methods and provide an up-to-date bibliography of the problem.

1. Introduction

In this survey, we discuss several algorithms for the factorization of univariate polynomials over finite fields. $\mathbb{F}_q$ denotes a finite field with q elements. We start by defining the problem.

> Given a monic univariate polynomial $f \in \mathbb{F}_q[x]$, find the complete factorization $f = f_1^{e_1} \cdots f_k^{e_k}$, where $f_1, \ldots, f_k$ are pairwise distinct monic irreducible polynomials and $e_1, \ldots, e_k$ are positive integers.

Finding the factorization of a polynomial over a finite field is of interest not only independently but also for many applications in computer algebra, algebraic coding theory, cryptography, and computational number theory. Polynomial factorization over finite fields is used as a subproblem in algorithms for factoring polynomials over the integers (Zassenhaus, 1969; Collins, 1979; Lenstra *et al.*, 1982; Knuth, 1998), for constructing cyclic redundancy codes and BCH codes (Berlekamp, 1968; MacWilliams and Sloane, 1977; van Lint, 1982), for designing public key cryptosystems (Chor and Rivest, 1985; Odlyzko, 1985; Lenstra, 1991), and for computing the number of points on elliptic curves (Buchmann, 1990).

Major improvements have been made in the polynomial factorization problem during this decade both in theory and in practice. From a theoretical point of view, asymptotically faster algorithms have been proposed. However, these advances are yet more striking in practice where variants of the asymptotically fastest algorithms allow us to factor polynomials over finite fields in reasonable amounts of time that were unassailable a few years ago. Our purpose in this survey is to stress the basic ideas behind these methods, to overview experimental results, as well as to give a comprehensive up-to-date bibliography of the problem. Kaltofen (1982, 1990, 1992) has given excellent surveys of

[§] E-mail: gathen@uni-paderborn.de
[¶] E-mail: daniel@cs.toronto.edu

0747–7171/01/010003 + 15 $35.00/0

the more general problem of factoring polynomials, while we only discuss univariate ones over finite fields. The scope of Bach and Shallit (1996) is even broader, but they do include a detailed account of our topic, as do the texts by Shparlinski (1999) and von zur Gathen and Gerhard (1999).

We organize this survey as follows. In Section 2, we present a general factoring algorithm. In Section 3, we discuss Berlekamp's algorithm. In Section 4, we summarize the probabilistic and deterministic algorithms that exist for factoring polynomials over finite fields. The general goal is to develop algorithms with running time bounded by a polynomial in the input size, that is, the degree of the polynomial to be factored and the logarithm of q. All results are given for asymptotic worst-case behaviour. We also briefly mention average-case analysis for polynomial factorization algorithms.

2. A General Factoring Algorithm

We assume that arithmetic in $\mathbb{F}_q$ is given. The cost measure of an algorithm will be the number of operations in $\mathbb{F}_q$, and sometimes we will use the "soft O" notation to ignore logarithmic factors: $g = O\tilde{}(n)$ means that $g = O(n(\log n)^l)$ for some constant l.

Polynomial factoring algorithms use basic polynomial operations such as products, divisions, gcd, powers of one polynomial modulo another, etc. A multiplication of two polynomials of degree at most n can be done in $O(n^2)$ operations in $\mathbb{F}_q$ using "classical" arithmetic, or in $O(n \log n \operatorname{loglog} n)$ operations in $\mathbb{F}_q$ using "fast" arithmetic (Schönhage and Strassen, 1971; Schönhage, 1977; Cantor, 1989; Cantor and Kaltofen, 1991; von zur Gathen and Gerhard, 1996). A division with remainder can be performed within the same time bounds. The cost of a gcd between two polynomials of degree at most n can be taken as $O(n^2)$ operations in $\mathbb{F}_q$ using classical methods, or as $O(n \log^2 n \operatorname{loglog} n)$ operations in $\mathbb{F}_q$ using fast methods (Aho *et al.*, 1974, Section 8.9). For polynomials h, g of degree at most n, the exponentiation $h^q \bmod g$ can be done by means of the classical *repeated squaring* method (see Knuth, 1998, pp. 461–462), with $O(\log q)$ polynomial products, i.e. $O(n^2 \log q)$ operations in $\mathbb{F}_q$ using classical methods, or $O(n \log q \log n \operatorname{loglog} n)$ operations in $\mathbb{F}_q$ using fast methods.

For a long time, it was widely believed that fast polynomial arithmetic was not practical for computer algebra problems; however, Shoup (1993) showed that this is not true. Indeed, his experiments give a crossover for the superiority of fast arithmetic already at polynomials of degree 25 modulo a 100-bit prime (see also Shoup, 1995). However, no comparison between fast methods and Karatsuba's algorithm (see Karatsuba and Ofman, 1962) seems to have been done for a general field $\mathbb{F}_q$ (for comparisons over $\mathbb{F}_2$, see Reischert 1995 and von zur Gathen and Gerhard 1996).

Let ω be an achievable exponent for matrix multiplication, so that we can multiply two $n \times n$ matrices with $O(n^\omega)$ operations in $\mathbb{F}_q$. Then systems of linear equations can be solved in $O(n^\omega)$ operations in $\mathbb{F}_q$. Classical linear algebra methods yield $\omega = 3$, and the current record is $\omega < 2.376$ (Coppersmith and Winograd, 1990).

Many (but not all) algorithms for factoring polynomials over finite fields comprise the following three stages:

SFF *squarefree factorization* replaces a polynomial by squarefree ones which contain all the irreducible factors of the original polynomial with exponents reduced to 1;

DDF *distinct-degree factorization* splits a squarefree polynomial into a product of polynomials whose irreducible factors all have the same degree;

EDF *equal-degree factorization* factors a polynomial whose irreducible factors have the same degree.

The algorithms for the first and second part are deterministic, while the fastest algorithms for the third part are probabilistic.

2.1. SQUAREFREE FACTORIZATION

Some factoring algorithms require that the input polynomials have no repeated factors. A polynomial $f \in \mathbb{F}_q[x]$ is *squarefree* if and only if for any $h, g \in \mathbb{F}_q[x]$ with $f = gh^2$ we have $h \in \mathbb{F}_q$. Thus, a polynomial is squarefree if it has no proper square divisors. If f is not squarefree, a factor can be found quickly by computing $\gcd(f, f^{'})$. In addition, we can find the *squarefree factorization* of a polynomial f of degree n, i.e. monic squarefree pairwise relatively prime polynomials $g_1, \ldots, g_k \in \mathbb{F}_q[x]$ such that $f = g_1 g_2^2 \cdots g_k^k$ and $g_k \neq 1$. Thus, g_i is the product of those monic irreducible polynomials in $\mathbb{F}_q[x]$ that divide f exactly to the power i.

An algorithm for finding the squarefree factorization of a polynomial is given below, with time $O^{\sim}(n^2 + n \log q)$. See Yun (1976) and Knuth (1998), Exercise 4.6, 2–36, for a method with running time $O(n \log^2 n \operatorname{loglog} n + n \log q)$ or $O^{\sim}(n \log q)$. Both in theory and practice, we can consider it a trivial step.

ALGORITHM Squarefree factorization (SFF).
Input: A monic polynomial $f \in \mathbb{F}_q[x]$ of degree $n \geq 1$, where char $(\mathbb{F}_q) = p$.
Output: A list of pairs (g_i, e_i), where g_i is a monic squarefree polynomial and e_i is an integer, such that $f = \prod g_i^{e_i}$.

$u := \gcd(f, f^{'})$;
`if` $u = 1$, `then return` $(f; 1)$;
`if` $1 \leq \deg u < n$, `then recursively compute SFF(`u`) and SFF(`f/u`).`
`Merge the output lists, and return the merged list.`
`if` $u = f$, `then` $f := \sum_{0 \leq i \leq n/p} f_{ip} x^{ip}$ `with` $f_0, f_p, \ldots, f_n \in \mathbb{F}_q$.
`Compute` $a_{ip} = f_{ip}^{1/p}$ `for all` i.
`Set` $h := \sum_{0 \leq i \leq n/p} a_{ip} x^i$, `and compute SFF(`$h$`) with output` $(h_1, d_1), \ldots, (h_l, d_l)$.
`Scale this by a factor of` p `and return SFF(`h^p`).`

To recover the squarefree factorization it is enough to collect factors with the same number of repetitions.

2.2. DISTINCT-DEGREE FACTORIZATION

The second step of the general factorization method is to find the *distinct-degree factorization*, that is, to split a squarefree polynomial into polynomials whose irreducible factors all have the same degree. Let $f \in \mathbb{F}_q[x]$ of degree n be the polynomial to be factored. The algorithm below is based on the following fact (see Lidl and Niederreiter, 1997, p. 91, Theorem 3.20).

Fact 2.1. *For $i \geq 1$, the polynomial $x^{q^i} - x \in \mathbb{F}_q[x]$ is the product of all monic irreducible polynomials in $\mathbb{F}_q[x]$ whose degree divides i.*

Algorithm Distinct-degree factorization (DDF).
Input: A monic squarefree polynomial $f \in \mathbb{F}_q[x]$ of degree n.
Output: The set of all pairs (g, d), where g is the product of all monic irreducible factors of f of degree d, with $g \neq 1$.

```
i := 1;   S := ∅;   f* := f;
while deg f* ≥ 2i do
      g := gcd(f*, x^(q^i) − x mod f*);
      if g ≠ 1, then S := S ∪ {(g, i)};
      f* := f*/g;
      i := i + 1;
endwhile;
if f* ≠ 1, then S := S ∪ {(f*, deg f*)};
return S.
```

The correctness of the above algorithm follows from Fact 2.1. The number of operations in $\mathbb{F}_q$ is $O\tilde{}(n^2 \log q)$ using the repeated squaring method, with a space requirement of $O(n)$ elements in $\mathbb{F}_q$. This algorithm was found by Gauß around 1798 and appears in his Nachlaß. It was rediscovered several times (Galois 1830 without explicitly mentioning the removal of factors; Serret 1866; Arwin 1918; Cantor and Zassenhaus 1981). The special case $d = 1$ was given by Legendre (1785).

The computation of the required powers can be improved using the "iterated Frobenius" method (von zur Gathen and Shoup, 1992, Algorithm 3.1). If $R = \mathbb{F}_q[x]/(f)$, the *Frobenius map* on R is defined by

$$\begin{aligned} \Phi : R &\longrightarrow R, \\ \alpha &\longmapsto \alpha^q. \end{aligned}$$

Computing iterates $\alpha, \alpha^q, \ldots, \alpha^{q^n}$ of the Frobenius map for $\alpha \in R$ is a basic component for distinct-degree factorization and several other problems in finite fields. Given $\alpha, \alpha^q, \ldots, \alpha^{q^m}$ by their canonical representatives $h_0, \ldots, h_m \in \mathbb{F}_q[x]$ of degree less than n, the iterated Frobenius method obtains the next m values $\alpha^{q^{m+1}}, \ldots, \alpha^{q^{2m}}$ using a fast multipoint evaluation algorithm to compute $h_m(\alpha^{q^i}) = \alpha^{q^{m+i}}$ for $1 \leq i \leq m$. Using the iterated Frobenius method, the above algorithm for distinct-degree factorization has running time $O\tilde{}(n^2 + n \log q)$, with a space requirement of $O(n^2)$ elements in $\mathbb{F}_q$.

In a typical distinct-degree factorization, most of the gcds computed are trivial. In order to reduce the number of these gcd's, von zur Gathen and Shoup (1992, Section 6), introduced a distinct-degree factorization based on the following blocking strategy. They divide the interval $1, \ldots, n$ into about $\sqrt{n}$ intervals of size $\sqrt{n}$, and then for each interval, they compute the joint product of the irreducible factors whose degree lies in that interval. A complete distinct-degree factorization is obtained with the help of the distinct-degree algorithm for each interval that contains at least two irreducible factors, and iterated Frobenius for computing powers. The running time of this algorithm is still $O\tilde{}(n^2 + n \log q)$, but the space requirement drops to $O(n\sqrt{n})$.

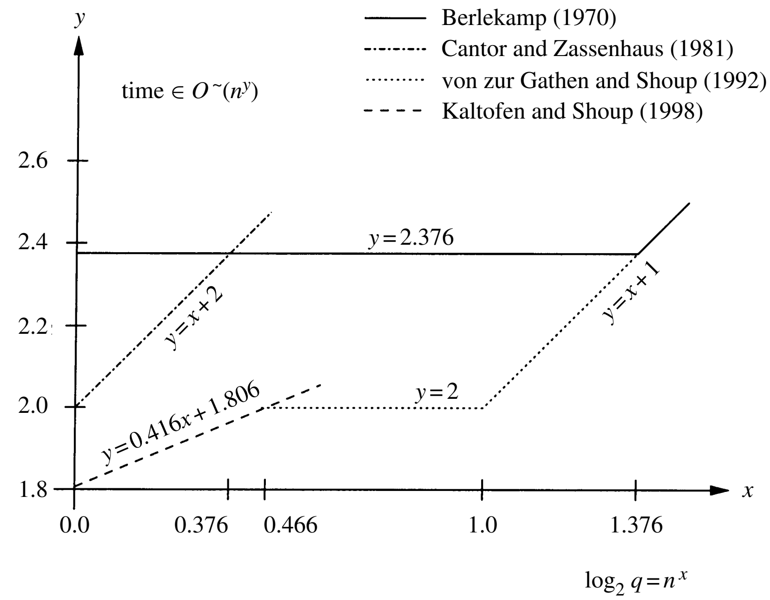

Figure 1. Running times of some factoring algorithms.

A new distinct-degree factorization algorithm is given in Kaltofen and Shoup (1998). They present a family of algorithms for this stage using fast matrix multiplication, and parametrized by β with $0 \leq \beta \leq 1$, that uses

$$O(n^{(\omega+1)/2+(1-\beta)(w-1)/2} + n^{1+\beta+o(1)} \log q)$$

operations in $\mathbb{F}_q$. Taking $\omega = 2.376$ and minimizing the exponent of n, they get an algorithm that uses $O(n^{1.815}(\log q)^{0.407})$ operations in $\mathbb{F}_q$. This is the first subquadratic-time algorithm for the distinct-degree factorization step, for small q. Since the algorithm uses fast matrix multiplication, its practicality is not clear, but they show how to adapt their technique to derive a practical version of this algorithm; see also Shoup (1995).

The problem is described in Figure 1 (essentially from Kaltofen and Shoup, 1998). The asymptotically fastest algorithms are compared considering the dependence between n and $\log q$, and using a fast matrix multiplication method. It shows the asymptotically fastest methods to be as follows: Kaltofen and Shoup (1998) for $\log q < n^{0.4545}$, von zur Gathen and Shoup (1992) for $n^{0.4545} < \log q < n^{1.376}$, and Berlekamp (1970) for larger fields. An algorithm by Huang and Pan (1998) beats the others when $\log q < n^{0.00173}$, with running time about $n^{1.80535} \log q$. Their method was also used in deriving the bound $0.416x + 1.806$ in Figure 1 which is not explicit in Kaltofen and Shoup (1998); see von zur Gathen and Gerhard (1999), Notes 14.8.

Major advances have occurred in the distinct-degree factorization problem over recent years not only in theory but also in practice. Shoup (1995) presents an implementation of a practical version of the algorithm in Kaltofen and Shoup (1998) using $O(n^{2.5} + n^{1+o(1)} \log q)$ operations in $\mathbb{F}_q$, with a space requirement of $O(n^{1.5})$ elements in $\mathbb{F}_q$. In addition, Shoup proposes a set of benchmarks for polynomial factorization algorithms consisting of factoring polynomials of degree n over an n-bit prime finite field (see also von zur Gathen 1992 and Monagan 1993). For instance, Shoup's algorithm factored a polynomial of degree 2048 modulo a 2048-bit prime in 12 days on a Sparc-10 workstation; the input size is about 0.5 MB.

An implementation over $\mathbb{F}_2$ of a variant of the distinct-degree factorization algorithm is presented in von zur Gathen and Gerhard (1996). As an example of the capability of this algorithm, it took two days to completely factor a pseudorandom polynomial of degree 262143 over $\mathbb{F}_2$ on two SPARC Ultra 1 workstations.

2.3. Equal-degree factorization

We concentrate on algorithms for factoring a monic squarefree univariate polynomial f over a finite field $\mathbb{F}_q$ of degree n with $r \geq 2$ irreducible factors $f_1, \ldots, f_r$, each of degree d. The algorithms that we present here are probabilistic.

First, we describe the algorithm in Cantor and Zassenhaus (1981). Since $f_1, \ldots, f_r$ are pairwise relatively prime, the Chinese Remainder Theorem provides the isomorphism:

$$\begin{aligned} \chi : \mathbb{F}_q[x]/(f) &\longrightarrow \mathbb{F}_q[x]/(f_1) \times \cdots \times \mathbb{F}_q[x]/(f_r), \\ h \bmod f &\longmapsto (h \bmod f_1, \ldots, h \bmod f_r). \end{aligned}$$

Let us write $R = \mathbb{F}_q[x]/(f)$, and $R_i = \mathbb{F}_q[x]/(f_i)$ for $1 \leq i \leq r$. Then R_i is a field with q^d elements and so contains $\mathbb{F}_q$

$$\mathbb{F}_q \subseteq \mathbb{F}_q[x]/(f_i) = R_i \cong \mathbb{F}_{q^d} \qquad \text{for } 1 \leq i \leq r.$$

Now f_i divides $h \in \mathbb{F}_q[x]$ if and only if $h \equiv 0 \bmod f_i$, that is, if and only if the ith component of $\chi(h \bmod f)$ is zero. Thus if $h \in \mathbb{F}_q[x]$ is such that $(h \bmod f_1, \ldots, h \bmod f_r)$ has some zero components and some nonzero components, i.e. $h \bmod f$ is a nonzero zerodivisor in R, then $\gcd(h, f)$ is a nontrivial factor of f, and we call h a "splitting polynomial". Therefore, we look for polynomials with this property.

First we consider odd q. We take $m = (q^d - 1)/2$ and an r-tuple $(h_1, \ldots, h_r)$ with each $h_i \in R_i^\times = \mathbb{F}_{q^d}^\times = \mathbb{F}_{q^d} \setminus \{0\}$. In $\mathbb{F}_{q^d}^\times$, half of the values are quadratic residues and the other half are quadratic nonresidues. Thus, $h_i^m = \pm 1$, with the same probability for both values when h_i is chosen randomly. Now, choose at random (uniformly) a polynomial $h \in \mathbb{F}_q[x]$, with $\deg h < n$, and let us assume that $\gcd(h, f) = 1$ (otherwise we have already found a partial factorization). The components $(h_1, \ldots, h_r)$ of its image under the Chinese remainder isomorphism are independently and uniformly distributed random elements in $R_i^\times = \mathbb{F}_{q^d}^\times$. Since $h_i^m = 1$ with probability $\frac{1}{2}$, the probability that $\gcd(h^m - 1, f)$ is not a proper factor of f, i.e. all the components in $(h_1^m - 1, \ldots, h_r^m - 1)$ are equal, is $2 \cdot 2^{-r} = 2^{-r+1} \leq \frac{1}{2}$. Running the algorithm l times ensures a probability of failure at most 2^{-l}.

After producing a factorization $f = g_1 g_2$, one may proceed in two ways: either by applying the algorithm recursively to g_1 and g_2, or by "refining" an already calculated factorization $f = \prod_{i=1}^{s} u_i$ by $\gcd(u_i, h^m - 1)$ for random h. For any $0 < \varepsilon < 1$, with $2\lceil \log \frac{r^2}{\varepsilon} \rceil$ such random choices, one obtains the complete factorization of f with probability at least $1 - \varepsilon$.

Algorithm Equal-degree factorization (EDF).
Input: $d \in \mathbb{N}$, a monic squarefree polynomial $f \in \mathbb{F}_q[x]$ of degree $n = rd$, with $r \geq 2$ irreducible factors each of degree d, and a confidence parameter ε.
Output: The set of monic irreducible factors of f, or "failure".

$\mathit{Factors} := \{f\}$; $k := 1$; $t := 2\lceil \log \frac{r^2}{\varepsilon} \rceil$;
`while` $k \le t$ `do`
  `Choose` $h \in \mathbb{F}_q[x]$ `with` $\deg h < n$ `at random;`
  $g := \gcd(h, f)$;
  `if` $g = 1$, `then`
    $g := h^{(q^d-1)/2} - 1 \pmod f$
  `endif;`
  `for each` $u \in \mathit{Factors}$ `with` $\deg u > d$ `do`
    `if` $\gcd(g,u) \ne 1$ `and` $\gcd(g,u) \ne u$, `then`
      $\mathit{Factors} := \mathit{Factors} \setminus \{u\} \cup \{\gcd(g,u), u/\gcd(g,u)\}$;
    `endif;`
  `if` $\mathit{Size}(\mathit{Factors}) = r$, `then return` $\mathit{Factors}$;
  $k := k + 1$;
`endwhile;`
`return 'failure'.`

The running time of the algorithm is $O\tilde{}(n^2 \log q \log^{-1} \varepsilon)$, and "failure" occurs with probability at most ε. By general principles, this can be turned into an algorithm that is guaranteed to factor f completely and whose running time is a random variable with mean $O\tilde{}(n^2 \log q)$ and exponentially decaying tails. For the special problem of finding roots, the idea of this algorithm can already be found in Legendre (1785, Section 28), but then was forgotten for almost two centuries.

Another probabilistic algorithm for equal-degree factorization is due to Ben-Or (1981) (see also Rabin 1980 and von zur Gathen and Shoup 1992, Section 3, Algorithm 3.6). These algorithms are based on trace computations of random elements in $R = \mathbb{F}_q[x]/(f)$. We choose $h \in R$ at random and compute its trace $g = \sum_{i=0}^{d-1} h^{q^i}$. The trace function has image $(\mathbb{F}_q)^r$, so raising g to the $(q-1)/2$ power in the case of odd characteristic, or computing $\sum_{i=0}^{k-1} g^{2^i}$ when $q = 2^k$, leads to a nontrivial factorization of f in a similar way, and with similar probabilities, as in Cantor and Zassenhaus (1981).

The running time of Ben-Or's algorithm is the same as that of Cantor and Zassenhaus (1981). A variant in the procedure for computing traces leads to the asymptotically fastest algorithm for the equal-degree problem, with $O\tilde{}(n^{(\omega+1)/2} + n \log q)$ operations in $\mathbb{F}_q$ (von zur Gathen and Shoup, 1992, Section 5).

Kaltofen and Shoup (1997) present new progress in the case of a sufficiently large extension $\mathbb{F}_q$ of its prime field $\mathbb{F}_p$, with $q = p^k$. If $k = n^x$ with $x > 1$, they improve the previous methods to

$$O\tilde{}(n^{2+x} + n^{1+1.69x} + n^{1+x} \log p)$$

operations in $\mathbb{F}_p$; here 1.69 represents $(\omega+1)/2$. When $k = \lceil n^{1.5} \rceil$ and $p = 2$, this gives $O(n(\log q)^{1.69})$ bit operations compared with previous $O(n(\log q)^2)$ bit operations. They also improve equal degree factorization, obtaining, e.g. $O(n^{2.69})$ bit operations for $q = 2^n$, compared with previous $O\tilde{}(n^3)$. Furthermore, they improve the picture of Figure 1 for a large field $\mathbb{F}_{2^k}$ with $k = n^x$ and $x > 1$, with the particular convention that $\log q$ bit operations count as much as one $\mathbb{F}_q$-operation. Their algorithm uses

$$O(n^{\max\{2, x(\omega-1)/2+1\}})$$

such operations, so that then the line $y = 2$ extends to $x = 2/(\omega - 1) \simeq 1.45$, and then rises as $y = \frac{\omega-1}{2}x + 1 \sim 0.69x + 1$.

As things now stand, equal-degree factorization, using randomized algorithms, can be done faster than distinct-degree factorization.

3. Algorithms Based on Linear Algebra

The pioneering modern algorithms for the factorization of polynomials over finite fields are from Berlekamp (1967, 1968, 1970). Let $f \in \mathbb{F}_q[x]$ be a monic squarefree univariate polynomial of degree n, and $f_1, \dots, f_r \in \mathbb{F}_q[x]$ its irreducible monic factors that we want to compute. If $R = \mathbb{F}_q[x]/(f)$ and $R_i = \mathbb{F}_q[x]/(f_i)$ for $1 \le i \le r$, then $R \cong R_1 \times \cdots \times R_r$ by the isomorphism of the Chinese Remainder Theorem. Recall from Section 2.2 the Frobenius map Φ on R. From Fermat's Little Theorem it follows that Φ is $\mathbb{F}_q$-linear. Consider the fixed points of Φ, i.e. the kernel of the mapping $\Phi - I$, where I is the identity function from R to itself. As in Camion (1980), we call $B = \{h \in R : \; h^q = h\}$ the *Berlekamp algebra*. For $a \in \mathbb{F}_{q^n}$, we have $a^q = a$ if and only if $a \in \mathbb{F}_q$, and thus $B \cong \mathbb{F}_q \times \cdots \times \mathbb{F}_q = (\mathbb{F}_q)^r$. The equality $h^q - h = \prod_{\alpha \in \mathbb{F}_q} (h - \alpha)$ for $h \in \mathbb{F}_q[x]$ implies that

$$f = \prod_{\alpha \in \mathbb{F}_q} \gcd(f, h - \alpha)$$

for $h \in \mathbb{F}_q[x]$ with $\overline{h} = (h \bmod f) \in B$. Berlekamp shows that when $\overline{h}$ runs through a basis for the r-dimensional $\mathbb{F}_q$ vector space B, then the common refinement of the resulting factorizations yields the complete factorization of f.

The running time of this algorithm is $O\tilde{}(n^3 + qn^2)$ operations in $\mathbb{F}_q$, using Gaussian elimination with classical arithmetic to find a basis of B.

Parts of the above method were known before Berlekamp. For instance, the above matrix construction appears in Petr (1937), Butler (1954) and Schwarz (1956), but it was Berlekamp who put all the elements together.

A randomized version of the above algorithm appears in Berlekamp (1970). He chooses v as a random element of B, and sets $u = v^{(q-1)/2} \bmod f$, assuming q to be odd (the even case can be treated with the trace function as in the equal-degree algorithm of Section 2). If $u \in B \backslash \mathbb{F}_q$, i.e. not all of the components of u are equal, then either $\gcd(f, u)$ or $\gcd(f, u-1)$ gives a nontrivial factor of f. As in the equal-degree factorization process, we have a probability of at least $\frac{1}{2}$ having a nontrivial factor of f. Therefore, the expected running time is $O\tilde{}(n^3 + n \log q)$ operations in $\mathbb{F}_q$.

For problems of large size, the basis computation in the Berlekamp algorithm can be done faster using Wiedemann's sparse linear system solver (Wiedemann, 1986). Kaltofen (1992) improves the running time of the algorithm in Berlekamp (1970) to $O\tilde{}(n^2 \log q)$ in this way, and Kaltofen and Lobo (1994) give an implementation of Berlekamp's algorithm that runs in $O\tilde{}(n^2 + n \log q)$ arithmetic operations. They use randomization and a Wiedemann parallel block linear system solver (Wiedemann, 1986; Coppersmith, 1994; Kaltofen, 1995) for finding nonzero elements of $\ker(\Phi - I)$. In 1993, this algorithm factored polynomials of degree 10001 over $\mathbb{F}_{127}$ in less than 4 days. The network used was composed of eight Sun 4 workstations with 32 Mbytes of memory each.

Another deterministic algorithm, also based on linear algebra, is presented in Niederreiter (1993a,b). Let $f \in \mathbb{F}_p[x]$ with $\deg f = n$ be the polynomial to be factored and $h \in \mathbb{F}_p[x]$ an unknown polynomial with degree less than n. Niederreiter's method is based on the system of n linear equations corresponding to the differential equation

$$f^p(h/f)^{(p-1)} + h^p = 0,$$

where the coefficients of h are the unknowns. Then h is used to factor f. Comparisons with Berlekamp's algorithm are in Miller (1992); Fleischmann (1993); Lee and Vanstone (1995), and Gao and von zur Gathen (1994); the latter paper applies Wiedemann's approach to this method. Roelse's (1999) parallel implementation of Niederreiter's algorithm can factor polynomials over $\mathbb{F}_2$ of degree 300000 in about three hours on an IBM SP-2 with 256 RS6000 processors.

4. Polynomial Factorization Algorithms

In this section, we give a list of some factoring algorithms from Berlekamp (1967) to the present. Berlekamp was the first to give a general algorithm for the problem. Some results prior to Berlekamp can be found in Lidl and Niederreiter (1997, at the end of Chapter 4), and Bach and Shallit (1996, pp. 195–198), give a substantial overview of the literature.

4.1. probabilistic algorithms

Berlekamp's 1970 paper was a pioneering result on probabilistic algorithms, whose huge success only took off later, after the work of Solovay and Strassen (1977) and Gill (1977). Today, probabilistic choice is used routinely in the many algorithmic applications where it is profitable.

In Sections 2 and 3, we presented the main ideas of the algorithms featured in the table below which lists those that gave an asymptotic improvement over previous results. In addition, other probabilistic algorithms for the problem are found in Calmet and Loos (1980); Lazard (1982) and Camion (1983b). For efficient factorization using fewer random bits see Bach and Shoup (1990). Finally, two recent randomized algorithms are in Gao and von zur Gathen (1994) and Kaltofen and Lobo (1994). An implementation of Cantor and Zassenhaus (1981) in Axiom is described in Naudin and Quitté (1998).

Probabilistic algorithms for the special problem of finding roots of polynomials over finite fields can be found in Berlekamp (1970); Rabin (1980); Ben-Or (1981), and van Oorschot and Vanstone (1989). The papers Rabin (1980) and Ben-Or (1981) also present algorithms for testing the irreducibility of polynomials over finite fields. The basic component of these algorithms is again Fact 2.1. Other probabilistic algorithms for testing the irreducibility of polynomials are in von zur Gathen and Shoup (1992) and Gao and Panario (1997). The fastest algorithm, with time $O\tilde{~}(n^2 + n \log q)$, to generate an irreducible polynomial is in Shoup (1994).

4.2. deterministic algorithms

Deterministic algorithms are the special case of probabilistic algorithms that make no use of their probabilistic choices. The first algorithm of this type is by Berlekamp (1967); see Section 3. Its running time is $O\tilde{~}(n^3 + qn^2)$, so it is not polynomial-time in $n \log q$. The major open problem in this area is to find a deterministic polynomial-time algorithm for the problem.

Deterministic algorithms are given in McEliece (1969); Camion (1983a); Menezes *et* *al.* (1988); Niederreiter (1993a,b); Rothstein and Zassenhaus (1994), and von zur Gathen and Shoup (1992, Section 9). The latter paper gives the currently fastest algorithm with $O\tilde{}(n^2 + n^{3/2}k + n^{3/2}k^{1/2}p^{1/2})$ operations in $\mathbb{F}_q$, where $q = p^k$. Shoup (1990b) gives a deterministic algorithm for the case of a prime field $\mathbb{F}_p$ with running time $O\tilde{}(n^2\sqrt{p})$, and Shparlinski (1999) gives an improvement. This algorithm factors "almost all" polynomials in polynomial time (see Shoup 1990b, Shparlinski 1999, Chapter 1, and Lange and Winterhof, 1999). Evdokimov (1994) gives an algorithm with time $(n^{\log n}\log q)^{O(1)}$, under the Extended Riemann Hypothesis (ERH); see also Gao (1999).

Under the ERH, deterministic polynomial-time factoring algorithms are known for some special cases. For the factorization of special polynomials, we have: Schoof (1985) for factoring quadratic polynomials over $\mathbb{F}_p$; Rónyai (1988) when $\deg f$ is small, Rónyai (1992) and Huang (1991a,b) for factoring polynomials whose Galois group (over $\mathbb{Q}$) is commutative such as for $x^n - a$ with $a \in \mathbb{Z}$. For special fields see: Moenck (1977); von zur Gathen (1987); Mignotte and Schnorr (1988) when $p - 1$ has only small prime factors, Bach *et al.* (1999) when some $\Phi_k(p)$ has this property. See Rónyai (1989a), Thiong Ly (1989), Shoup (1991a), Shoup (1991b), and Menezes *et al.* (1992) for related results.

4.3. Average-case analysis

Few results are known in terms of average-case analysis of polynomial factorization algorithms. Shoup (1990b) studies his deterministic algorithm using estimates for the number of solutions of equations over finite fields and Weil's bounds (for a similar analysis see Ben-Or 1981, and for background on these issues see Schmidt 1976).

Flajolet *et al.* (1996) present a complete average-case analysis of the general algorithm in Section 2; see also Panario *et al.* (1998). For this purpose, a study of the distribution of the degrees of irreducible factors, the probability of irreducibility, and so on, is needed. This involves counting polynomials over finite fields verifying special characteristics. Some of these results were known for random polynomials of large degree n over $\mathbb{F}_q$. Flajolet *et al.* (1996) give a unified study of parameters relevant to factoring polynomials over finite fields.

A simple variant of the distinct-degree factorization gives an irreducibility test for polynomials (Ben-Or, 1981). The average-case analysis of this algorithm is in Panario and Richmond (1998). Another irreducibility test due to Rabin (1980) is analyzed in Panario and Viola (1998). These analyses are based on a framework that includes generating functions to describe the parameters studied and asymptotic analysis to extract coefficients (see also Panario, 1997).

4.4. Further references

We list some works that were not cited in the text: Tonelli (1891), Schwarz (1939, 1940), Golomb *et al.* (1959), Prange (1959), Schwarz (1960, 1961), Lloyd (1964), Lloyd and Remmers (1966), Chien *et al.* (1969), Prešić (1970), Agou (1976a,b), Adleman *et al.* (1977), Agou (1977), Chen and Li (1977), Willett (1978), Agou (1980), Mignotte (1980), Camion (1981), Gunji and Arnon (1981), Camion (1982), Adleman and Lenstra (1986), Kaltofen (1987), Evdokimov (1989), Poli and Gennero (1989), Knopfmacher and Knopfmacher (1990), Lenstra (1990), Shoup (1990a), Wang (1990), Trevisan and Wang (1991), Evdokimov (1993), Knopfmacher and Knopfmacher (1993), Niederreiter and Göttfert (1993), Shparlinski (1993a,b), Davis (1994), Niederreiter (1994a,b), Knopfmacher (1995), Knopfmacher and Warlimont (1995), Niederreiter and Göttfert (1995), Fleischmann and Roelse (1996), Rónyai and Szánto (1996), Gao *et al.* (1999), Wan (1999).

References

Adleman, L. M., Lenstra, Jr., H. W. (1986). Finding irreducible polynomials over finite fields. In *Proc. 18th Ann. ACM Symp. on the Theory of Computing, Berkeley, CA*, pp. 350–355. New York, ACM Press.

Adleman, L. M., Manders, K., Miller, G. (1977). On taking roots in finite fields. In *Proc. 18th Ann. IEEE Symp. on Foundations of Computer Science, Providence, RI*, pp. 175–178. CA, U.S.A, IEEE Computer Society Press.

Agou, S. (1976a). Factorisation des polynômes à coefficients dans un corps fini. *Publ. Dep. Math., Lyon*, **13**, 63–71.

Agou, S. (1976b). Factorisation sur un corps fini k des polynômes composés $f(x^s)$ lorsque $f(x)$ est un polynôme irreductible de $k[x]$. *Compt. Rend. Acad. Sci.*, 1067–1068.

Agou, S. (1977). Factorisation sur un corps fini $\mathbb{F}_{p^n}$ des polynômes composés $f(x^{p^r} - ax)$ lorsque $f(x)$ est un polynôme irreductible de $\mathbb{F}_{p^n}[x]$. *J. Number Theory*, **9**, 229–239.

Agou, S. (1980). Sur la factorisation des polynômes $f(x^{p^{2r}} - ax^{p^r} - bx)$ sur un corps fini $\mathbb{F}_{p^s}$. *J. Number Theory*, **12**, 447–459.

Aho, A. V., Hopcroft, J. E., Ullman, J. D. (1974). *The Design and Analysis of Computer Algorithms*, Reading, MA, Addison-Wesley.

Arwin, A. (1918). Über Kongruenzen von dem fünften und höheren Graden nach einem Primzahlmodulus. *Ark. Mat. Astron. Fys.*, **14**, 1–46.

Bach, E., von zur Gathen, J., Lenstra, Jr., H. W. (1999). Deterministic factorization of polynomials over special finite fields. In *Finite Fields and Their Applications*, to appear.

Bach, E., Shallit, J. (1996). *Algorithmic Number Theory, Volume 1: Efficient Algorithms*, Cambridge, MA, MIT Press.

Bach, E., Shoup, V. (1990). Factoring polynomials using fewer random bits. *J. Symb. Comput.*, **9**, 229–239.

Ben-Or, M. (1981). Probabilistic algorithms in finite fields. In *Proc. 22nd Ann. IEEE Symp. on Foundations of Computer Science, Nashville, TN*, pp. 394–398. CA, U.S.A, IEEE Computer Society Press.

Berlekamp, E. R. (1967). Factoring polynomials over finite fields. *Bell Syst. Tech. J.*, **46**, 1853–1859.

Berlekamp, E. R. (1968). *Algebraic Coding Theory*, New York, McGraw-Hill.

Berlekamp, E. R. (1970). Factoring polynomials over large finite fields. *Math. Comput.*, **24**, 713–735.

Buchmann, J. (1990). Complexity of algorithms in algebraic number theory. In *Number Theory. Proc. First Conf. Canadian Number Theory Association*, pp. 37–53. Walter de Gruyter.

Butler, M. C. R. (1954). On the reducibility of polynomials over a finite field. *Q. J. Math. Oxford*, **5**, 102–107.

Calmet, J., Loos, R. (1980). An improvement of Rabin's probabilistic algorithm for generating irreducible polynomials over $GF(p)$. *Inf. Process. Lett.*, **11**, 94–95.

Camion, P. (1980). Un algorithme de construction des idempotents primitifs d'idéaux d'algèbres sur $\mathbb{F}_q$. *Compt. Rend. Acad. Sci. Paris*, **291**, 479–482.

Camion, P. (1981). Factorisation des polynômes de $\mathbb{F}_q$. *Rev. Cethedec*, **18**, 1–17.

Camion, P. (1982). Un algorithme de construction des idempotents primitifs d'idéaux d'algèbres sur $\mathbb{F}_q$. *Ann. Discrete Math.*, **12**, 55–63.

Camion, P. (1983a). A deterministic algorithm for factorizing polynomials of $\mathbb{F}_q[X]$. *Ann. Discrete Math.*, **17**, 149–157.

Camion, P. (1983b). Improving an algorithm for factoring polynomials over a finite field and constructing large irreducible polynomials. *IEEE Trans. Inf. Theory*, **IT-29**, 378–385.

Cantor, D. G. (1989). On arithmetical algorithms over finite fields. *J. Comb. Theory, A*, **50**, 285–300.

Cantor, D. G., Kaltofen, E. (1991). On fast multiplication of polynomials over arbitrary algebras. *Acta Inform.*, **28**, 693–701.

Cantor, D. G., Zassenhaus, H. (1981). A new algorithm for factoring polynomials over finite fields. *Math. Comput.*, **36**, 587–592.

Chen, J., Li, X. (1977). Using the $Q[f]$ matrix to determine the structure of polynomials over finite fields [in Chinese]. *Acta Math. Sin.*, **20**, 294–297.

Chien, R. T., Cunningham, B. D., Oldham, I. B. (1969). Hybrid methods for finding roots of a polynomial—with application to BCH decoding. *IEEE Trans. Inf. Theory*, **IT-15**, 329–335.

Chor, B., Rivest, R. L. (1985). A knapsack type public key cryptosystem based on arithmetic in finite fields. *IEEE Trans. Inf. Theory*, **IT-34**, 901–909.

Collins, G. E. (1979). Factoring univariate integral polynomials in polynomial average time. In *Proceedings of EUROSAM '79, Marseille, France,* LNCS **72**, pp. 317–329. Springer-Verlag.

Coppersmith, D. (1994). Solving homogeneous linear equations over GF(2) via block Wiedemann algorithm. *Math. Comput.*, **62**, 333–350.
Coppersmith, D., Winograd, S. (1990). Matrix multiplication via arithmetic progressions. *J. Symb. Comput.*, **9**, 251–280.
Davis, R. A. (1994). Idempotent computations over finite fields. *J. Symb. Comput.*, **17**, 237–258.
Evdokimov, S. A. (1989). Factoring a solvable polynomial over a finite field and the Generalized Riemann Hypothesis, Zapiski Nauchnyck Seminarov Leningradskgo Otdeleniya Mat. Inst. V. A. Steklova Akad. Nauk SSSR (LOMI). **176**, 104–117. With English abstract.
Evdokimov, S. A. (1993). Efficient factorization of polynomials over finite fields and the generalized Riemann hypothesis. Technical Report, Universität Bonn.
Evdokimov, S. A. (1994). Factorization of polynomials over finite fields in subexponential time under GRH. In *Proceedings of the First International ANTS Symposium*, LNCS **877**, pp. 209–219. Springer-Verlag.
Flajolet, P., Gourdon, X., Panario, D. (1996). Random polynomials and polynomial factorization. In Meyer auf der Heide, F., Monien, B. eds, *Proc. 23rd Int. Colloq. Automata, Languages and Programming ICALP 1996, Paderborn, Germany*, LNCS **1099**, pp. 232–243. Springer-Verlag, See also, INRIA Rapport de Recherche No 3370, March 1998, 28 pages: http://pavillac.inria.fr/algo/Flajolet/Publications/RR3370.ps.
Fleischmann, P. (1993). Connections between the algorithms of Berlekamp and Niederreiter for factoring polynomials over $\mathbb{F}_q$. *Linear Algebr. Appl.*, **192**, 101–108.
Fleischmann, P., Roelse, P. (1996). Comparative implementations of Berlekamp's and Niederreiter's polynomial factorization algorithms. In Cohen, S. ed., *Finite Fields and Applications, Proceedings of the Third International Conference, Glasgow*, LNS **233**, pp. 73–83.
Galois, É. (1830). Sur la théorie des nombres. *Bull. Sci. Math. Férussac*, **13**, 428–435. See also (1846). *J. Math. Pures Appl.*, **11**, 398–407. and In Bourgne, R., Azra, J.-P. eds, (1962). *Écrits et mémoires d'Évariste Galois*, pp. 112–128. Paris, Gauthier-Villars.
Gao, S. (1999). On the deterministic complexity of factoring polynomials. *J. Symb. Comput.*, **31**, 19–36. doi:10.1006/jsco.1999.1001.
Gao, S., von zur Gathen, J. (1994). Berlekamp's and Niederreiter's polynomial factorization algorithms. In Mullen, G. L., Shiue, P. J.-S. eds, *Finite Fields: Theory, Applications and Algorithms*, volume 168 of Contemporary Mathematics, pp. 101–115. Brookline, MA, American Mathematical Society.
Gao, S., Howell, J., Panario, D. (1999). Irreducible polynomials of given forms. In Mullin, R. C., Mullen, G. L. eds, *Finite Fields: Theory, Applications and Algorithms*, volume 225 of Contemporary Mathematics, pp. 43–54. Brookline, MA, American Mathematical Society.
Gao, S., Panario, D. (1997). Tests and constructions of irreducible polynomials over finite fields. In Cucker, F., Shub, M. eds, *Foundations of Computational Mathematics*, pp. 346–361. Springer Verlag.
von zur Gathen, J. (1987). Factoring polynomials and primitive elements for special primes. *Theor. Comput. Sci.*, **52**, 77–89.
von zur Gathen, J. (1992). A polynomial factorization challenge. *ACM SIGSAM Bull.*, **26**, 22–24.
von zur Gathen, J., Gerhard, J. (1996). Arithmetic and factorization of polynomials over $\mathbb{F}_2$. In Lakshman, Y. N. ed., *Proc. 1996 Int. Symp. on Symbolic and Algebraic Computation ISSAC '96, Zürich, Switzerland*, pp. 1–9. ACM Press. Technical Report tr-rsfb-96-018, University of Paderborn, 1996, 43 pp, `http://www-math.uni-paderborn.de /~aggathen/Publications/polyfactTR.ps`.
von zur Gathen, J., Gerhard, J. (1999). *Modern Computer Algebra,* xiv + 754 pp, Cambridge, UK, Cambridge University Press.
von zur Gathen, J., Shoup, V. (1992). Computing Frobenius maps and factoring polynomials. *Comput. Complexity*, **2**, 187–224.
Gill, J. (1977). Computational complexity of probabilistic Turing machines. *SIAM J. Comput.*, **6**, 675–695.
Golomb, S. W., Welch, L. R., Hales, A. (1959). On the factorization of trinomials over GF(2). Jet Propulsion Laboratory memo, 20–189.
Gunji, H., Arnon, D. (1981). On polynomial factorization over finite fields. *Math. Comput.*, **36**, 281–287.
Huang, M. A. (1991a). Factorization of polynomials over finite fields and decomposition of primes in algebraic number fields. *J. Algorithms*, **12**, 464–481.
Huang, M. A. (1991b). Generalized Riemann hypothesis and factoring polynomials over finite fields. *J. Algorithms*, **12**, 482–489.
Huang, X., Pan, V. Y. (1998). Fast rectangular matrix multiplication and applications. *J. Complexity*, **14**, 257–299.
Kaltofen, E. (1982). Factorization of polynomials. In Buchberger, B., Collins, G. E., Loos, R. eds, *Computer Algebra, Symbolic and Algebraic Computation*, 2nd edn, pp. 95–113. New York, Springer-Verlag.
Kaltofen, E. (1987). Deterministic irreducibility testing of polynomials over large finite fields. *J. Symb. Comput.*, **4**, 77–82.

Kaltofen, E. (1990). Polynomial factorization 1982–1986. In Chudnovsky, D. V., Jenks, R. D. eds, *Computers in Mathematics*, pp. 285–309. New York, Marcel Dekker, Inc.

Kaltofen, E. (1992). Polynomial factorization 1987–1991. In Simon, I. ed., *Proceedings of LATIN '92, São Paulo, Brazil*, LNCS **583**, pp. 294–313. Springer-Verlag.

Kaltofen, E. (1995). Analysis of Coppersmith's block Wiedemann algorithm for the parallel solution of sparse linear systems. *Math. Comput.*, **64**, 777–806.

Kaltofen, E., Lobo, A. (1994). Factoring high-degree polynomials by the black box Berlekamp algorithm. In von zur Gathen, J., Giesbrecht, M. eds, *Proceedings of ISSAC'94*, pp. 90–98. New York, ACM Press.

Kaltofen, E., Shoup, V. (1997). Fast polynomial factorization over high algebraic extensions of finite fields. In Küchlin, W. W. ed., *Proc. 1997 Int. Symp. on Symbolic and Algebraic Computation ISSAC '97, Maui*, pp. 184–188. New York, ACM Press.

Kaltofen, E., Shoup, V. (1998). Subquadratic-time factoring of polynomials over finite fields. *Math. Comput.*, **67**, 1179–1197.

Karatsuba, A., Ofman, Y. (1963). Multiplication of multidigit numbers on automata. *Sov. Phys.–Doklady*, **7**, 595–596.

Knopfmacher, A. (1995). Enumerating basic properties of polynomials over a finite field. *S. Af. J. Sci.*, **91**, 10–11.

Knopfmacher, A., Knopfmacher, J. (1990). Counting polynomials with a given number of zeros in a finite field. *Linear and Multilinear Algebra*, **26**, 287–292.

Knopfmacher, A., Knopfmacher, J. (1993). Counting irreducible factors of polynomials over a finite field. *Discrete Math.*, **112**, 103–118.

Knopfmacher, A., Warlimont, R. (1995). Distinct degree factorizations for polynomials over a finite field. *Trans. Am. Math. Soc.*, **347**, 2235–2243.

Knuth, D. E. (1998). *The Art of Computer Programming, Volume 2, Seminumerical Algorithms*, 3^{rd} edn. First edition 1969., Reading, MA, Addison-Wesley.

Lange, T., Winterhof, A. (1999). Factoring polynomials over arbitrary finite fields. *Theor. Comput. Sci.*, to appear

Lazard, D. (1982). On polynomial factorization. In *Proceedings of EUROCAL 1982*, LNCS **144**, pp. 126–134. Springer-Verlag.

Lee, T., Vanstone, S. (1995). Subspaces and polynomial factorization over finite fields. *Appl. Algebr. Eng., Commun. Comput.*, **6**, 147–157.

Legendre, A. M. (1785). Recherches d'analyse indéterminée. *Mém. Acad. R. Sci.*, 465–559.

Lenstra, A. K., Lenstra, Jr., H. W., Lovász, L. (1982). Factoring polynomials with rational coefficients. *Math. Ann.*, **261**, 515–534.

Lenstra, Jr., H. W. (1990). Algorithms for finite fields. In Loxton, J. H. ed., *Number Theory and Cryptography*, LMSLN **154**, pp. 76–85. Cambridge, UK, Cambridge University Press.

Lenstra, Jr., H. W. (1991). On the Chor-Rivest knapsack cryptosystem. *J. Cryptol.*, **3**, 149–155.

Lidl, R., Niederreiter, H. (1997). *Finite Fields*, volume 20 of Encyclopedia of Mathematics and its Applications. Cambridge, U.K., Cambridge University Press, 2nd edn. First edition 1983, published by Addision-Wesley, Reading, MA, U.S.A.

van Lint, J. H. (1982). *Introduction to Coding Theory*, volume 86 of Graduate Texts in Mathematics. New York, Springer-Verlag.

Lloyd, D. B. (1964). Factorization of the general polynomial by means of its homomorphic congruential functions. *The American Mathematical Monthly*, **71**, 863–870.

Lloyd, D. B., Remmers, H. (1966). Polynomial factor tables over finite fields. *Math. Algorithms*, **1**, 85–99.

MacWilliams, F. J., Sloane, N. J. A. (1977). *The Theory of Error-Correcting Codes*, volume 16 of Mathematical Library. Amsterdam, North-Holland.

McEliece, R. J. (1969). Factorization of polynomials over finite fields. *Math. Comput.*, **23**, 861–867.

Menezes, A. J., van Oorschot, P. C., Vanstone, S. A. (1988). Some computational aspects of root finding in GF(q^m). In Gianni, P. ed., *Proc. 1988 Int. Symp. on Symbolic and Algebraic Computation ISSAC '88,Rome, Italy*, LNCS **358**, pp. 259–270. Springer-Verlag.

Menezes, A. J., van Oorschot, P. C., Vanstone, S. A. (1992). Subgroup refinement algorithms for root finding in $GF(q)$. *SIAM J. Comput.*, **21**, 228–239.

Mignotte, M. (1980). Calcul des racines d-ièmes dans un corps fini. *Compt. Rend. Acad. Sci. Paris*, **290**, 205–206.

Mignotte, M., Schnorr, C. (1988). Calcul déterministe des racines d'un polynôme dans un corps fini. *Compt. Rend. Acad. Sci. Paris*, **306**, 467–472.

Miller, V. S. (1992). On the factorization method of Niederreiter. Preprint.

Moenck, R. T. (1977). On the efficiency of algorithms for polynomial factoring. *Math. Comput.*, **31 (137)**, 235–250.

Monagan, M. B. (1993). von zur Gathen's factorization challenge. *ACM SIGSAM Bull.*, **20**, 13–18.

Naudin, P., Quitté, C. (1998). Univariate polynomial factorization over finite fields. *Theor. Comput. Sci.*, **191**, 1–36.

Niederreiter, H. (1993a). Factorization of polynomials and some linear algebra problems over finite fields. *Linear Algebr. Appl.*, **192**, 301–328.
Niederreiter, H. (1993b). A new efficient factorization algorithm for polynomials over small finite fields. *Appl. Algebr. Eng., Commun. Comput.*, **4**, 81–87.
Niederreiter, H. (1994a). Factoring polynomials over finite fields using differential equations and normal bases. *Math. Comput.*, **62**, 819–830.
Niederreiter, H. (1994b). New deterministic factorization algorithms for polynomials over finite fields. In Mullen, G. L., Shiue, P. J.-S. eds, *Finite Fields: Theory, Applications and Algorithms*, volume 168 of Contemporary Mathematics, pp. 251–268. Brookline, MA, American Mathematical Society.
Niederreiter, H., Göttfert, R. (1993). Factorization of polynomials over finite fields and characteristic sequences. *J. Symb. Comput.*, **16**, 401–412.
Niederreiter, H., Göttfert, R. (1995). On a new factorization algorithm for polynomials over finite fields. *Math. Comput.*, **64**, 347–353.
Odlyzko, A. (1985). Discrete logarithms and their cryptographic significance. In *Advances in Cryptology: Proceedings of EUROCRYPT 1984, Paris, France,* LNCS **209**, pp. 224–314. Springer-Verlag.
van Oorschot, P., Vanstone, S. (1989). A geometric approach to root finding in GF(q^m). *IEEE Trans. Inf. Theory*, **IT-35**, 444–453.
Panario, D. (1997). Combinatorial and algebraic aspects of polynomials over finite fields. Ph. D. Thesis, Department of Computer Science, University of Toronto,Technical Report 306/97, 154 pp.
Panario, D., Gourdon, X., Flajolet, P. (1998). An analytic approach to smooth polynomials over finite fields. In Buhler, J. P. ed., *Algorithmic Number Theory, Proceedings of ANTS-III,* LNCS **1423**, pp. 226–236. Portland, OR, Springer-Verlag.
Panario, D., Richmond, B. (1998). Analysis of Ben-Or's polynomial irreducibility test. *Random Struct. Algorithms*, **13**, 439–456.
Panario, D., Viola, A. (1998). Analysis of Rabin's polynomial irreducibility test. In Lucchesi, C. L., Moura, A. V. eds, *Proceedings of LATIN '98, Campinas, Brazil*, LNCS **1380**, pp. 1–10. Springer-Verlag.
Petr, K. (1937). Über die Reduzibilität eines Polynoms mit ganzzahligen Koeffizienten nach einem Primzahlmodul. *Časopis pro pěstovánímatematiky a fysiky*, **66**, 85–94.
Poli, A., Gennero, M. C. (1989). Fast16: A software program for factorising polynomials over large $GF(p)$. In Huguet, L., Poli, A. eds, *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes: AAECC-5, Menorca, Spain*, LNCS **356**, pp. 139–156. Springer-Verlag.
Prange, E. (1959). An algorithm for factoring $x^n - 1$ over a finite field. Technical Report AFCRC-TN-59-775, Air Force Cambridge Research Center, Bedford MA.
Prešić, M. D. (1970). A method for solving equations in finite fields. *Mat. Vesnik*, **7**, 507–509.
Rabin, M. O. (1980). Probabilistic algorithms in finite fields. *SIAM J. Comput.*, **9**, 273–280.
Reischert, D. (1995). Schnelle Multiplikation von Polynomen über GF(2) und Anwendungen. Diplomarbeit. Institut für Informatik II, Rheinische Friedrich-Wilhelm-Universität Bonn, Germany.
Roelse, P. (1999). Factoring high–degree polynomials over $\mathbb{F}_2$ with Niederreiter's algorithm on the IBM SP2. *Math. Comput.*, **68,** 869–880.
Rónyai, L. (1988). Factoring polynomials over finite fields. *J. Algorithms*, **9**, 391–400.
Rónyai, L. (1989a). Factoring polynomials modulo special primes. *Combinatorica*, **9**, 199–206.
Rónyai, L. (1992). Galois groups and factoring polynomials over finite fields. *SIAM J. Discrete Math.*, **5**, 345–365.
Rónyai, L., Szánto, A. (1996). Prime-field-complete functions and factoring polynomials over finite fields. *Comput. Artificial Intelligence*, **6**, 571–577.
Rothstein, M., Zassenhaus, H. (1994). Deterministic analysis of aleatoric methods of polynomial factorization over finite fields. *J. Number Theory*, **47**, 20–42.
Schmidt, W. (1976). *Equations over Finite Fields*, LNM **536**. New York, Springer-Verlag.
Schönhage, A. (1977). Schnelle Multiplikation von Polynomen über Körpern der Charakteristik 2. *Acta Inform.*, **7**, 395–398.
Schönhage, A., Strassen, V. (1971). Schnelle Multiplikation großer Zahlen. *Computing*, **7**, 281–292.
Schoof, R. J. (1985). Elliptic curves over finite fields and the computation of square roots mod p. *Math. Comput.*, **44**, 483–494.
Schwarz, Š. (1939). *Contribution à la réductibilité des polynômes dans la théorie des congruences*, Věstník Královské České Společnosti Nauk, Třída Matemat.-Přirodověd Ročník Praha, pp. 1–7.
Schwarz, Š. (1940). Sur le nombre des racines et des facteurs irréductibles d'une congruence donnée. *Časopis pro pěstovánímatematiky a fysiky*, **69**, 128–145.
Schwarz, Š. (1956). On the reducibility of polynomials over a finite field. *Q. J. Math. Oxford*, **7**, 110–124.
Schwarz, Š. (1960). On a class of polynomials over a finite field. *Matematicko-Fyzikálny Časopis*, **10**, 68–80.
Schwarz, Š. (1961). On the number of irreducible factors of a polynomial over a finite field. *Czech. Math. J.*, **11**, 213–225.
Serret, J.-A. (1866). *Cours d'algèbre supérieure*, 3rd edn. Paris, Gauthier-Villars.

Shoup, V. (1990a). New algorithms for finding irreducible polynomials in finite fields. *Math. Comput.*, **54**, 435–447.

Shoup, V. (1990b). On the deterministic complexity of factoring polynomials over finite fields. *Inf. Process. Lett.*, **33**, 261–267.

Shoup, V. (1991a). A fast deterministic algorithm for factoring polynomials over finite fields of small characteristic. In Watt, S. M. ed., *Proc. 1991 Int. Symp. on Symb. and Algebraic Computation ISSAC '91, Bonn, Germany*, pp. 14–20. ACM Press.

Shoup, V. (1991b). Smoothness and factoring polynomials over finite fields. *Inf. Process. Lett.*, **38**, 39–42.

Shoup, V. (1993). Factoring polynomials over finite fields: asymptotic complexity vs. reality. In *Proc. Int. IMACS Symp. on Symbolic Computation, New Trends and Developments, Lille, France*, pp. 124–129.

Shoup, V. (1994). Fast construction of irreducible polynomials over finite fields. *J. Symb. Comput.*, **17**, 371–391.

Shoup, V. (1995). A new polynomial factorization algorithm and its implementation. *J. Symb. Comput.*, **20**, 363–397.

Shparlinski, I. E. (1993a). Finding irreducible and primitive polynomials. *Appl. Algebr. Eng., Commun. Comput.*, **4**, 263–268.

Shparlinski, I. E. (1993b). On bivariate polynomial factorization over finite fields. *Math. Comput.*, **60**, 787–791.

Shparlinski, I. E. (1999). *Finite Fields: Theory and Computation*, volume 477 of Mathematics and its Applications. Kluwer Academic Publishers.

Solovay, R., Strassen, V. (1977). A fast Monte-Carlo test for primality. *SIAM J. Comput.*, **6**, 84–85. Erratum in (1978). **7**, 118.

Thiong Ly, A. (1989). A deterministic algorithm for factorizing polynomials over extensions GF(p^m) of GF(p), p a small prime. *J. Inf. Optim. Sci.*, **10**, 337–344.

Tonelli, A. (1891). *Bemerkung über die Auflösung quadratischer Congrenzen*, Göttinger Nachrichten, pp. 344–346.

Trevisan, V., Wang, P. (1991). Practical factorization of univariate polynomials over finite fields. In Watt, S. M. ed., *Proc. 1991 Int. Symp. on Symbolic and Algebraic Computation ISSAC '91, Bonn, Germany*, pp. 22–31. ACM Press.

Wan, D. (1999). Computing zeta functions over finite fields. In Mullin, R. C., Mullen, G. L. eds, *Finite Fields: Theory, Applications and Algorithms*, volume 225 of Contemporary Mathematics, pp. 131–141. Brookline, MA, American Mathematical Society.

Wang, P. S. (1990). Parallel univariate polynomial factorization on shared-memory multiprocessors. In Watanabe, S., Nagata, M. eds, *Proc. 1990 Int. Symp. on Symbolic and Algebraic Computation ISSAC '90, Tokyo, Japan*, pp. 145–151. ACM Press.

Wiedemann, D. H. (1986). Solving sparse linear equations over finite fields. *IEEE Trans. Inf. Theory*, **IT-32**, 54–62.

Willett, M. (1978). Factoring polynomials over a finite field. *SIAM J. Appl. Math.*, **35**, 333–337.

Yun, D. Y. Y. (1976). On square-free decomposition algorithms. In Jenks, R. D. ed., *Proc. 1976 ACM Symp. on Symbolic and Algebraic Computation ISSAC '76, Yorktown Heights, NY*, pp. 26–35. New York, ACM Press.

Zassenhaus, H. (1969). On Hensel factorization, I. *J. Number Theory*, **1**, 291–311.

Originally Received 1 August 1996
Accepted 12 March 1999