Algorithms for Factoring Square-Free Polynomials over Finite Fields

Chelsea Richards

August 7, 2009

Given a polynomial in GF$(q)[x]$, there are simple and well known algorithms for determining its square-free part. Assuming that $a(x)$ is a monic, square-free polynomial of degree n, we will present four algorithms for determining its complete factorization. For what follows let $q = p^m$ where p is a prime number and GF(q) is a finite field consisting of q elements.

8.4 Berlekamp's Factorization Algorithm

The first algorithm we consider is due to Berlekamp. In order to explain Berlekamp's method, we need to introduce some related concepts and notation. First let V be the ring of residue classes given by GF$(q)[x]/ < a(x) >$. It is clear that V is a vector space over GF(q) which is generated by the set $\{1, x, x^2, \ldots, x^{n-1}\}$ and is therefore of dimension n. Now identify the set $\{[v(x)] \in \text{V} : [v(x)]^q = [v(x)]\}$ with W$= \{v(x) \in \text{GF}(q)[x] : v(x)^q \equiv v(x) \bmod a(x)\}$ and the residue class $[v(x)]$ with $v(x) \bmod a(x)$.

Theorem 8.4: The subset W is also a subspace of the vector space V.

The proof of this theorem is an easy exercise using the definition of a subspace and the following:
Let $a, b \in \text{GF}(q)$. Then

$$(a+b)^q = a^q + \tbinom{q}{1}a^{q-1}b + \tbinom{q}{2}a^{q-2}b^2 + \ldots + \tbinom{q}{q-1}ab^{q-1} + b^q.$$

And since q divides $\binom{q}{k}$ for any $k = 1 \ldots (q-1)$,

$$(a+b)^q = a^q + b^q.$$

Lemma 8.1:(Fermat's little theorem) For any $r \in \mathrm{GF}(q)$, $r^q = r$.

Proof: Clearly $0^q = 0$. So, let r be a nonzero element of $\mathrm{GF}(q)$. Then the set $\{1, r, r^2, \ldots\}$ is a finite subgroup of the multiplicative group of $\mathrm{GF}(q)$, which has order $q-1$. By Lagrange's theorem, the order ρ, of the subgroup $\{1, r, r^2, \ldots\}$ and hence of r must be such that $\rho \mid q-1$. This implies that $r^{q-1} = 1$ and hence $r^q = r$, for any nonzero $r \in \mathrm{GF}(q)$. $\square$

Lemma 8.2: Suppose $a(x)$ is irreducible in $\mathrm{GF}(q)[x]$. Then W is a subspace of dimension one in V.

Proof: Since $a(x)$ is irreducible we know that V is a field. Therefore $p(z) = z^q - z \in \mathrm{V}[x]$ can have at most q roots. By Lemma 8.1 (Fermat's little theorem), $r^q = r$ for all $r \in \mathrm{GF}(q)$, so each of the q distinct elements of $\mathrm{GF}(q)$ satisfy $r^q - r = 0$. Hence we have that all q roots of $p(z)$ are constants in $\mathrm{GF}(q)$. That is, W consists of q constant polynomials and therefore can be identified with $\mathrm{GF}(q)$, and generated by the single element $\{1\}$. So W is a subspace of dimension one in V. $\square$

Theorem 8.5: The dimension of W is equal to the number of irreducible factors of $a(x)$.

Proof: Let $a(x) = a_1(x)a_2(x)\ldots a_k(x)$ be the unique monic, irreducible factorization of $a(x)$. For each i from 1 to k let $\mathrm{V}_i = \mathrm{GF}(q)[x]/ < a_i(x) >$. By the Chinese remainder theorem the mapping

$$\phi : \mathrm{V} \longrightarrow \mathrm{V}_1 \times \mathrm{V}_2 \times \cdots \times \mathrm{V}_k$$

defined by $\phi(v(x)) = (v(x) \bmod a_1(x), v(x) \bmod a_2(x), \ldots, v(x) \bmod a_k(x))$ is a ring isomorphism. Now since $v^q \equiv v \bmod a(x)$ implies that $v^q \equiv v \bmod a_i$ for each i from 1 to k, ϕ induces the ring homomorphism

$$\phi_{\mathrm{W}} : \mathrm{W} \longrightarrow \mathrm{W}_1 \times \mathrm{W}_2 \times \ldots \times \mathrm{W}_k$$

where $\mathrm{W}_i = \{s \in \mathrm{V}_i : s^q = s\}$. From Lemma 8.2, since each $a_i(x)$ is irreducible, each V_i is a field and each W_i can be identified with $\mathrm{GF}(q)$. Now to see that the dimension of W is k, we need that ϕ_{W} is an isomorphism. Consider $(s_1, s_2, \ldots, s_k) \in \mathrm{W}_1 \times \mathrm{W}_2 \times \cdots \times \mathrm{W}_k$. Since ϕ is onto, there exists a $v(x) \in$V such that $\phi(v(x)) = (s_1, s_2, \ldots, s_k)$. Then we know

$$\phi(v(x)^q) = (s_1^q, s_2^q, \ldots, s_k^q) = (s_1, s_2, \ldots, s_k) = \phi(v(x)),$$

which, since ϕ is an isomorphism (and therefore one-to-one), implies that $v(x)^q = v(x)$. That is, $v(x)$ is in W, and hence ϕ_{W} is onto. That ϕ_{W} is

one-to-one follows directly from the fact that ϕ has this property. So ϕ_{W} is an isomorphism. Since each of the W_i has dimension one this means that W then has dimension k, which is the number of irreducible factors of $a(x)$. $\square$

Now given W, we can determine the number of irreducible factors of the polynomial. For this to be useful we need to be able to calculate W, and then, in order to factor the polynomial we need a method for determining the factors once we know W. First we will look at how to find the factors assuming that W is known.

Theorem 8.6: Let $a(x)$ be a monic, square-free polynomial in $\mathrm{GF}(q)[x]$ and let $v(x)$ be a nonconstant polynomial in W. Then

$$a(x) = \prod_{s \in \mathrm{GF}(q)} \mathrm{GCD}(v(x) - s, a(x)).$$

Proof: In $\mathrm{GF}(q)[x]$

$$x^q - x = \prod_{s \in \mathrm{GF}(q)} (x - s).$$

Therefore

$$v(x)^q - v(x) = \prod_{s \in \mathrm{GF}(q)} (v(x) - s).$$

Since $v(x) \in \mathrm{W}$, $a(x)$ divides $v(x)^q - v(x)$ and hence for all $i = 1 \ldots k$,

$$a_i(x) \mid (v(x)^q - v(x)) = \prod_{s \in \mathrm{GF}(q)} (v(x) - s).$$

Now since $\mathrm{GCD}(v(x) - s, v(x) - t) = 1$ for all $s \neq t$, for a given i, $a_i(x)$ must divide $v(x) - s_i$ for exactly one $s_i \in \mathrm{GF}(q)$. Therefore for $i = 1 \ldots k$,

$$a_i(x) \mid \mathrm{GCD}(a(x), v(x) - s_i)$$

and so

$$a(x) \mid \prod_{i=1}^{k} \mathrm{GCD}(a(x), v(x) - s_i) \mid \prod_{s \in \mathrm{GF}(q)} \mathrm{GCD}(a(x), v(x) - s).$$

Clearly $\mathrm{GCD}(a(x), v(x) - s) \mid a(x)$ for each s, and since all the $v(x) - s$ are relatively prime for distinct s, this implies that

$$\prod_{s \in \mathrm{GF}(q)} \mathrm{GCD}(a(x), v(x) - s) \mid a(x).$$

And hence

$$a(x) = \prod_{s \in \mathrm{GF}(q)} \mathrm{GCD}(a(x), v(x) - s). \square$$

Now we have a method for determining the factors of $a(x)$ assuming that we know the elements of W. Since W is a vector space, it is enough to have a basis for W. So the next step is to find a way to calculate a basis for the vector space W.

For any polynomial $v(x) \in \mathrm{GF}(q)[x]$, we have

$$\begin{aligned} v(x)^q &= (v_0 + v_1 x + \ldots + v_{n-1} x^{n-1})^q \\ &= v_0^q + v_1^q x^q + \ldots + v_{n-1} x^{q(n-1)} \text{ by Theorem 8.4} \\ &= v_0 + v_1 x^q + \ldots + v_{n-1} x^{q(n-1)} = v(x^q) \text{ by Lemma 8.1.} \end{aligned}$$

Therefore

$$\begin{aligned} \mathrm{W} &= \{v(x) \in \mathrm{GF}(q)[x] : v(x)^q - v(x) = 0 \bmod a(x)\} \\ &= \{v(x) \in \mathrm{GF}(q)[x] : v(x^q) - v(x) = 0 \bmod a(x)\}. \end{aligned}$$

This description of W represents the vector space as the solution space to a system of n equations in n unknowns. The coefficient matrix of this system is the $n \times n$ matrix Q whose entries $q_{i,j}$ (for $0 \le i, j \le n-1$) are determined by

$$x^{q \cdot j} = q_{0,j} + q_{1,j} x + \ldots + q_{n-1,j} x^{n-1} \bmod a(x).$$

That is, the matrix Q has columns $1 \ldots n$ determined from the remainders of division of $x^0, x^q, x^{q \cdot 2}, \ldots, x^{q \cdot (n-1)}$, respectively, by $a(x)$.

Theorem 8.7: Given W and Q as defined previously,

$$\mathrm{W} = \{\mathbf{v} = (v_0, \ldots, v_{n-1}) : (Q - I) \cdot \mathbf{v} = \mathbf{0}\}.$$

Proof: The equation

$$v(x^q) - v(x) \equiv 0 \bmod a(x)$$

is equivalent to

$$\begin{aligned} 0 &\equiv \sum_{j=0}^{n-1} v_j x^{q \cdot j} - \sum_{j=0}^{n-1} v_j x^j \bmod a(x) \\ &\equiv \sum_{j=0}^{n-1} v_j [\sum_{i=0}^{n-1} q_{i,j} \cdot x^i] - \sum_{j=0}^{n-1} v_j x^j \bmod a(x) \\ &\equiv \sum_{i=0}^{n-1} \{\sum_{j=0}^{n-1} v_j \cdot q_{i,j} - v_i\} \cdot x^i \bmod a(x). \end{aligned}$$

And since the coefficients of the x^i must all be zero, this implies

$$\sum_{j=0}^{n-1} v_j \cdot q_{i,j} - v_i \equiv 0$$

for all $i = 0 \ldots n-1$, which is equivalent to

$$Q \cdot (v_0, \ldots, v_{n-1}) - (v_0, \ldots, v_{n-1}) = (0, \ldots, 0)$$

which is equivalent to

$$(Q - I) \cdot \mathbf{v} = \mathbf{0}.$$

Since W consists of all $v(x)$ satisfying $v(x^q) - v(x) \equiv 0 \bmod a(x)$, which is equivalent to $(Q-I)\cdot \mathbf{v} = \mathbf{0}$, then W consists of all $\mathbf{v}$ satisfying the latter. □

That W is the null space of the matrix $Q - I$ is a consequence of Theorem 8.7. Thus, in order to determine a basis for W, we need only use linear algebra to calculate a basis for the null space of $Q - I$. Now it remains only to describe a method for computing the Q matrix for a complete description of Berlekamp's algorithm.

Generating the Q matrix involves calculating

$$x^q \bmod a(x), x^{2q} \bmod a(x), \ldots, x^{(n-1)q} \bmod a(x).$$

This can be done by using an iterative procedure that generates $x^{m+1} \bmod a(x)$ given that $x^m \bmod a(x)$ has been determined. If

$$a(x) = a_0 + a_1 x + \ldots + a_{n-1} x^{n-1} + x^n$$

and

$$x^m = c_{m,0} + c_{m,1} x + \ldots + c_{m,n-1} x^{n-1} \bmod a(x)$$

then working modulo $a(x)$ we have

$$\begin{aligned} x^{m+1} &\equiv c_{m,0}x + c_{m,1}x^2 + \ldots + c_{m,n-1}x^n \\ &\equiv c_{m,0}x + c_{m,1}x^2 + \ldots + c_{m,n-1}(-a_0 - a_1x - \ldots - a_{n-1}x^{n-1}) \\ &\equiv -c_{m,n-1}a_0 + (c_{m,0} - c_{m,n-1}a_1)x + \ldots + (c_{m,n-2} - c_{m,n-1}a_{n-1})x^{n-1} \\ &\equiv c_{m+1,0} + c_{m+1,1}x + \ldots + c_{m+1,n-1}x^{n-1} \end{aligned}$$

where

$$c_{m+1,0} = -c_{m,n-1}a_0$$

and

$$c_{m+1,i} = c_{m,i-1} - c_{m,n-1}a_i$$

for $i = 1 \ldots n-1$. Thus, we can generate the Q matrix by storing a vector $\mathbf{c}$ of elements from GF(q):

$$\mathbf{c} \leftarrow (c_0, \ldots, c_{n-1})$$

is initialized by

$$\mathbf{c} \leftarrow (1, 0, \ldots, 0)$$

and is updated by

$$\mathbf{c} \leftarrow (-c_{n-1} \cdot a_0, c_0 - c_{n-1} \cdot a_1, \ldots, c_{n-2} - c_{n-1} \cdot a_{n-1}).$$

Then after each $(iq)^{th}$ iteration, the entries of the vector are copied into the i^{th} column of the Q matrix. In this way, computing the Q matrix requires $q \cdot n$ multiplications for each new column and since Q has a total of n columns, generating the entire matrix is $\mathrm{O}(q \cdot n^2)$ operations in GF(q).

Algorithm 8.4: Form Q Matrix

```
procedure FormMatrixQ(a(x), q)
    #Given a polynomial a(x) of degree n in GF(q)[x], calculate
    #the Q matrix required by Berlekamp's algorithm
    n ← deg(a(x)); c ← (1, 0, ..., 0); Column(0, Q) ← c;
    for m from 1 to (n − 1)q do {
        c ← (−c_{n−1} · a_0, c_0 − c_{n−1} · a_1, ..., c_{n−2} − c_{n−1} · a_{n−1})
        if q | m then
            Column(m/q, Q) ← c }
    return (Q)
end
```

Now we have all the components necessary for Berlekamp's method. The algorithm first computes the Q matrix, then calculates the null space of $Q-I$ to obtain a basis for the set W. Then it breaks $a(x)$ into factors of lower degree by computing greatest common divisors of $a(x)$ and the difference of basis elements of W and elements of GF(q). This process is then applied repeatedly to further factor the factors, until the number of factors is equal to the dimension of W and hence a complete factorization has been determined.

Algorithm 8.5: Berlekamp's Factoring Algorithm

procedure Berlekamp($a(x), q$)
#Given a monic, square-free polynomial $a(x) \in \mathrm{GF}(q)[x]$
#calculate irreducible factors $a_1(x), \ldots, a_k(x)$ such
#that $a(x) = a_1(x) \cdots a_k(x)$.
$Q \leftarrow$ FormMatrixQ($a(x), q$)
Compute $\{\mathbf{v}^{[1]}, \mathbf{v}^{[2]}, \ldots, \mathbf{v}^{[k]}\}$ a basis for the null space of $(Q - I)$
#Note we can ensure that $\mathbf{v}^{[1]} = (1, 0, \ldots, 0)$
$factors \leftarrow \{a(x)\}$
$r \leftarrow 2$
while SizeOf($factors$) $< k$ **do** {
foreach $u(x) \in factors$ **do** {
foreach $s \in \mathrm{GF}(q)$ **do** {
$g(x) \leftarrow \mathrm{GCD}(v^{[r]} - s, u(x))$
if $g(x) \neq 1$ **and** $g(x) \neq u(x)$ **then** {
$w(x) \leftarrow u(x)/g(x)$
$factors \leftarrow factors - \{u(x)\} \cup \{g(x), w(x)\}$ }
if SizeOf($factors$)$= k$ **then return**($factors$) }
$r \leftarrow r + 1$ }}
end

Example 8.6: Using Algorithm 8.4 we will find the irreducible factors of the polynomial

$$a(x) = x^6 - 3x^5 + x^4 - 3x^3 - x^2 - 3x + 1 \in \mathrm{GF}(11)[x].$$

The first step is to determine the Q matrix. Since $x^0 = 1 \equiv 1 \bmod a(x)$, Column 1 of Q is given by $[1, 0, \ldots, 0]^{\mathrm{T}}$. Then we multiply by x and mod out by $a(x)$ repeatedly until we have $x^{11} \bmod a(x)$ to get Column 2. So we compute that $x^{11} \bmod a(x) = 5x^5 - 5x^4 - 3x^3 - 3x^2 + 5x + 3$ and hence Column 2 of Q is $[3, 5, -3, -3, -5, 5]^{\mathrm{T}}$. Continuing in this way, we eventually obtain the entries of all 6 columns of Q, the last of which comes from the

coefficients of x^{55} mod $a(x)$. At the end of this we have the matrix:

$$Q = \begin{pmatrix} 1 & 3 & 3 & -2 & -4 & -3 \\ 0 & 5 & -5 & 4 & -3 & -1 \\ 0 & -3 & -5 & -1 & -1 & -4 \\ 0 & -3 & 1 & 3 & 0 & -3 \\ 0 & -5 & -1 & -4 & 0 & 1 \\ 0 & 5 & 0 & -2 & -3 & -3 \end{pmatrix}$$

Next, we need a basis for the null space of the matrix:

$$Q - I = \begin{pmatrix} 0 & 3 & 3 & -2 & -4 & -3 \\ 0 & 4 & -5 & 4 & -3 & -1 \\ 0 & -3 & 5 & -1 & -1 & -4 \\ 0 & -3 & 1 & 2 & 0 & -3 \\ 0 & -5 & -1 & -4 & -1 & 1 \\ 0 & 5 & 0 & -2 & -3 & -4 \end{pmatrix}$$

This is a matter of simple linear algebra and using Maple's Nullspace command in this case, we can see that the vectors

$$\mathbf{v}^{[1]} = (1,0,0,0,0,0), \mathbf{v}^{[2]} = (0,1,1,1,1,0), \mathbf{v}^{[3]} = (0,0,-4,-2,0,1)$$

form a basis for W. In terms of polynomial representation, this gives

$$v^{[1]}(x) = 1,\ v^{[2]}(x) = x^4 + x^3 + x^2 + x,\ v^{[3]}(x) = x^5 - 2x^3 - 4x^2.$$

Now we know that $a(x)$ has three irreducible factors, since the basis is formed by three vectors. To find the factorization we need to perform a series of GCD calculations. Starting with $v^{[2]}$, we take the GCD of the polynomial $a(x)$ with each s in GF(q) subtracted from the basis polynomial. In each case if the GCD is non-trivial, we re-assign the quotient of $a(x)$ divided by that GCD to $a(x)$ and continue. First we have

$$\mathrm{GCD}(a(x), v^{[2]} - 0) = x + 1.$$

Thus $a_1(x) = x + 1$ is one of the factors of $a(x)$. Now let

$$a(x) = \frac{a(x)}{x+1} = x^5 - 4x^4 + 5x^3 + 3x^2 - 4x + 1$$

and then calculate $\mathrm{GCD}(a(x), v^{[2]} - 1) = x^2 + 5x + 3$. Then

$$\frac{a(x)}{x^2 + 5x + 3} = x^3 + 2x^2 + 3x + 4,$$

which gives two other factors of $a(x)$, so we have the factorization

$$a(x) = (x+1)(x^2+5x+3)(x^3+2x^2+3x+4).$$

Note that if $\mathrm{GCD}(a(x), v^{[2]}(x) - s)$ had been trivial for all $s \in \mathbf{Z}_{11}$ then we would have repeated the above process with $v^{[3]}(x)$. Also if there had been non-trivial factors and if after the GCD calculations had been completed for all $s \in \mathbf{Z}_{11}$ the number of factors was still less than three, we would have repeated the process with each of the individual factors in place of $a(x)$. □

Theorem 8.8 : The cost of Berlekamp's algorithm for computing the factors of a monic, square-free polynomial $a(x)$ of degree n, which has k distinct irreducible factors, in the domain $\mathrm{GF}(q)$ is $\mathrm{O}(k \cdot q \cdot n^2 + n^3)$.

Proof: As we saw previously, generating the Q matrix costs $\mathrm{O}(q \cdot n^2)$ field operations. Determining a basis for W involves Gaussian elimination of the $n \times n$ matrix $Q - I$, which costs $\mathrm{O}(n^3)$ field multiplications. Then, in the algorithm, each of the k factors requires q GCD calculations, each at an approximate cost of n^2 operations. Therefore, this last step involves approximately $k \cdot q \cdot n^2$ field operations, leading to a total cost of $\mathrm{O}(k \cdot q \cdot n^2 + n^3)$ field operations. □

Having established the cost of this algorithm we can see that for large q the method becomes infeasible. In order to modify this into a viable procedure for large q, we need a more efficient means of generating the Q matrix, as well as a strategy to avoid calculating the GCD's for all $s \in \mathrm{GF}(q)$ exhaustively.

8.5 The Big Prime Berlekamp Algorithm

The efficiency of the Q matrix generation can be improved through the use of binary powering to compute large powers of x mod $a(x)$. Using this method, calculating Column 2 of the matrix, i.e. calculating x^q modulo $a(x)$, requires $\log(q)$ steps, each of which involves multiplication of two polynomials of degree less than n and division by $a(x)$, which is of degree n. Therefore, this column is determined in $\log(q) \cdot n^2$ operations, assuming classical algorithms for multiplication and division. To compute $x^q \bmod a(x), x^{2q} \bmod a(x), \ldots, x^{(n-1)q} \bmod a(x)$, we first calculate $h = x^q \bmod a(x)$ using binary powering. Then $x^{kq} = h \cdot x^{(k-1)q} \bmod a(x)$. In this way, each subsequent column requires $\mathrm{O}(n^2)$ further operations, and since there are n columns, the total cost of generating the Q matrix in this

way is $O(\log(q) \cdot n^2 + n^3)$.

Although using binary powering is a significant improvement, if we still require all the GCD calculations of the previous algorithm, the complexity of the overall algorithm still involves $O(k \cdot q \cdot n^2)$, which is impractical for large q. Zassenhaus came up with a first attempt to reduce the cost of the GCD step. He did this by first determining which $s \in \mathrm{GF}(q)$ would produce nontrivial GCD's for a given $v(x)$. To describe Zassenhaus's method, for a given $v(x)$ define

$$S = \{s \in \mathrm{GF}(q) : \mathrm{GCD}(v(x) - s, a(x)) \neq 1\}$$

and

$$m_v(x) = \prod_{s \in S} (x - s).$$

Theorem 8.9: The polynomial $m_v(x)$ as defined above is the minimal polynomial for $v(x)$. That is, $m_v(x)$ is the polynomial of least degree such that

$$m_v(v(x)) \equiv 0 \bmod a(x).$$

Proof: Choose an arbitrary factor $a_i(x)$ of $a(x)$. Then since

$$\mathrm{GCD}(v(x) - s, a(x)) \neq 1$$

for all $s \in S$, it must be that

$$a_i(x) \mid \mathrm{GCD}(v(x) - s, a(x))$$

for some $s \in S$. Hence $a_i(x) \mid (v(x) - s)$ for some $s \in S$. Then since this is true for arbitrary i,

$$a(x) \mid \prod_{s \in S} (v(x) - s) = m_v(v(x)).$$

Now to show that this is the polynomial of least degree with this property, suppose that it is not. That is, suppose $m(x)$ is a polynomial of smaller degree than $m_v(x)$ and that $m(v(x)) \equiv 0 \bmod a(x)$. Then there must exist an s in S such that

$$m(x) = q(x) \cdot (x - s) + r$$

where r is a nonzero constant in GF(q). Since s is in S, one of the factors of $a(x)$, say $a_i(x)$, divides $v(x) - s$. Then $a_i(x)$ also divides $m(v(x))$ since

$$m(v(x)) \equiv 0 \bmod a(x).$$

From $m(x) = q(x) \cdot (x - s) + r$, we have

$$m(v(x)) = q(v(x)) \cdot (v(x) - s) + r,$$

which implies that $a_i(x) \mid r$, since $a_i(x) \mid m(v(x))$ and $a_i(x) \mid (v(x) - s)$. But r is a constant, so this is a contradiction and hence $m_v(x)$ is the minimal polynomial. $\square$

There is a standard method for computing the minimal polynomial for a given $v(x)$, using linear algebra. Before describing how this is done, we look at how to factor the minimal polynomial $m(x)$ into its linear factors, i.e. to find the $s \in$ GF(q) which give nontrivial GCD's. To do this, we can use Rabin's probabilistic method for root finding. This method makes use of the factorization

$$x^q - x = x \cdot (x^{(q-1)/2} - 1) \cdot (x^{(q-1)/2} + 1)$$

for q odd. From Lemma 8.1 we know that $x^q - x$ is the product of all monic, linear polynomials in GF(q). So each linear factor of $m(x)$ that is not x divides either $x^{(q-1)/2} - 1$ or $x^{(q-1)/2} + 1$. If the factors are distributed in such a way that at least one divides each, then we can split $m(x)$ by taking GCD$(m(x), x^{(q-1)/2} - 1)$. However, it may be that the factors are not distributed in this way and therefore the GCD is trivial. Rabin showed that if we shift x by α, for a random $\alpha \in$ GF(q), then the probability that GCD$(m(x), (x - \alpha)^{(q-1)/2} - 1)$ is nontrivial is at least one half. We also want to compute this GCD for large q (i.e. $q >>$ deg(m)) without explicitly constructing $w(x) = (x - \alpha)^{(q-1)/2}$. To do this we use

$$\begin{aligned} \text{GCD}(m(x), w(x) - 1) &= \text{GCD}(m(x), (w(x) - 1) \bmod m(x)) \\ &= \text{GCD}(m(x), (w(x) \bmod m(x)) - 1) \end{aligned}$$

and compute $w(x)$ mod $m(x)$ using binary powering with remainder so that degrees never exceed $2 \cdot r$, where r is the degree of $m(x)$.

So to execute Rabin's method to factor $m(x)$, which we know is a product of linear factors, we select a constant $\alpha \in$ GF(q) at random, calculate $w(x) = (x - \alpha)^{(q-1)/2}$ mod $m(x)$ using binary powering, and then compute

$\mathrm{GCD}(m(x), w(x) - 1)$. An average of two random α will need to be tried to get a nontrivial GCD. Once one has been found, we have two factors of $m(x)$, and to completely factor $m(x)$ into its linear factors we just continue applying this process to each of the factors until we find those that are irreducible, i.e. of degree one in this case. The cost of splitting $m(x)$ the first time is $\mathrm{O}(r^2 \cdot \log(q))$ since we raise $x - \alpha$ to the $(q-1)/2 \in O(q)$ modulo $m(x)$ which has degree r, and then take the GCD of two polynomials with maximum degree r, which costs $\mathrm{O}(r^2)$. The sum of the cost of the subsequent steps is approximately the same as the cost of the first so in fact $\mathrm{O}(r^2 \cdot \log(q))$ is the complexity of finding all of the roots of $m(x)$.

We can use Rabin's method to find all the linear factors of a general polynomial $a(x)$ by first computing the greatest common divisor of $m(x) = x^q - x \bmod a(x)$ and $a(x)$, which gives the product of the linear factors, and then applying the above description to $m(x)$.

Algorithm 8.6: Rabin's Root Finding Algorithm

procedure Rabin$(a(x), q)$
#Given a monic, square-free polynomial $a(x) \in \mathrm{GF}(q)[x]$,
#find all of the linear factors of $a(x)$.
$m(x) \leftarrow x^q - x \bmod a(x)$
$m(x) \leftarrow \mathrm{GCD}(m(x), a(x))$
$n \leftarrow \deg(m(x))$
if $n > 1$ **then**
$g(x) \leftarrow 1$
while $g(x) = 1$ **or** $g(x) = m(x)$ **do** {
Choose $\alpha \in \mathrm{GF}(q)$ at random
Calculate $w(x) = (x - \alpha)^{(q-1)/2}$ using binary powering
$g(x) \leftarrow \mathrm{GCD}(w(x) - 1, m(x))\}$
$m(x) \leftarrow m(x)/g(x)$
$factors \leftarrow \mathrm{Rabin}(m(x), q) \cup \mathrm{Rabin}(g(x), q)$
else
$factors \leftarrow \{m(x)\}$
return$(factors)$
end

Now we need the following method for computing the minimal polynomial for a given $v(x) \in \mathrm{W}$. For each integer t, starting at $t = 1$ and increasing, we determine $v(x)^t \bmod a(x)$. Calculating each new power of $v(x)$ involves multiplying by $v(x)$ modulo $a(x)$ which is of degree n. After

each new power of $v(x)$ is found, solve

$$m_0 + m_1 v(x) + \ldots + m_{t-1} v(x)^{t-1} + v(x)^t \equiv 0 \bmod a(x).$$

If $m(x)$ is of degree r then we will need to compute r powers of $v(x)$ in total, so that the cost of this step is $\mathrm{O}(r \cdot n^2)$. Then at each step we are solving an n by t system. This needs to be done carefully and incrementally, using information from our attempt to solve the previous system for each new t, so that the total cost of solving is equal to the cost of finding the first nontrivial solution. This is an n by r linear system, which costs $\mathrm{O}(r^2 \cdot n)$ field operations to solve. This nontrivial solution gives us $m(x)$, the minimal polynomial of degree r. In order to find all k factors of $a(x)$ using this method, we need that $r = k$ at this point. Therefore, if $r < k$ then choose a new $v(x)$ and repeat the above until a minimal polynomial with degree equal to the size of the basis for W is found. Then we factor $m(x)$, using Rabin's method, and of all the elements of $\mathrm{GF}(q)$, we need only compute the GCD's for the s which are roots of $m(x)$.

Algorithm 8.7: Zassenhaus's Minimal Polynomial Factoring Algorithm

procedure Zassenhaus($a(x), q$)
  #Given a monic, square-free polynomial $a(x) \in \mathrm{GF}(q)[x]$
  #calculate the irreducible factors using Zassenhaus's
  #minimal polynomial method with Rabin's root finding
  #algorithm to factor the minimal polynomial
  Compute the Q matrix using binary powering
  Calculate $\{\mathbf{v}^{[1]}, \mathbf{v}^{[2]}, \ldots, \mathbf{v}^{[k]}\}$ a basis for the null space of $Q - I$
  $m(x) \leftarrow 1$
  repeat
    Choose a nonconstant $v(x) \in \mathrm{W}$ at random
    Compute the minimal polynomial
      $m(x) = m_0 + m_1 x + \ldots + m_{r-1} x^{r-1} + x^r \in \mathrm{GF}(q)[x]$
      satisfying $m(v(x)) \equiv 0 \bmod a(x)$
  until degree($m(x)$) $= k$
  Factor $m(x)$ using Algorithm 8.6
  Take S to be the set of roots of $m(x)$
  $factors \leftarrow \{\}$
  foreach $s \in S$ do {
    $g(x) \leftarrow \mathrm{GCD}(v(x) - s, a(x))$
    $factors \leftarrow factors \cup \{g(x)\}$ }
  return($factors$)
end

Example 8.7: To demonstrate the method of using the minimal polynomial, we will apply it to the polynomial from Example 8.6

$$a(x) = x^6 - 3x^5 + x^4 - 3x^3 - x^2 - 3x + 1 \in \mathrm{GF}(11)[x].$$

Let $v(x) = v^{[2]}(x) = x^4 + x^3 + x^2 + x$, which we know is in W. To determine the minimal polynomial, first find

$$v(x)^2 \equiv -2x^5 + 2x^4 - 5x^3 - x^2 + 2x + 5 \bmod a(x).$$

Then solve

$$a + b \cdot v(x) + v(x)^2 \equiv 0 \bmod a(x)$$

and find that there are no nonzero solutions. So calculate

$$v(x)^3 \equiv x^5 - 5x^4 + 4x^3 + 2x^2 - 5x + 3 \bmod a(x).$$

Next set up and solve

$$a + b \cdot v(x) + c \cdot v(x)^2 + v(x)^3 \equiv 0 \bmod a(x)$$

and find that there is a nontrivial solution given by $a = 0, b = 4, c = -5$. Therefore the minimal polynomial is

$$m_v(x) = x^3 - 5x^2 + 4x,$$

which has degree $k = 3$. Now we factor $m_v(x)$ by first trying $v(x) = x - 5$. We calculate

$$\mathrm{GCD}(m_v(x) = x^3 - 5x^2 + 4x, (x-5)^5 - 1) = 1,$$

so this does not provide any information. Choosing another $v(x) = x - 1$, $\mathrm{GCD}(m_v(x), v(x)^5 - 1) = x - 4$. So we have one of the linear factors of $m_v(x)$. Now we need only factor $x^3 - 5x^2 + 4x/(x-4) = x^2 - x$. Selecting $v(x) = x - 2$ and computing $\mathrm{GCD}(x^2 - x, (x-2)^5 - 1) = x$, we see that x is another factor and therefore the third and final linear factor is $(x^2 - x)/x = x - 1$. Hence

$$m_v(x) = x \cdot (x-1) \cdot (x-4)$$

and we know that when applying the GCD calculations for Berlekamp's algorithm, we need only check those $s \in S = \{0, 1, 4\}$.
So now

$$g(x) = \mathrm{GCD}(a(x), v(x)) = x + 1$$

is one of the factors and what remains is

$$u(x) = a(x)/g(x) = x^5 - 4x^4 + 5x^3 + 3x^2 - 4x + 1.$$

Then take

$$g(x) = \mathrm{GCD}(u(x), v(x) - 1) = x^2 + 5x + 3.$$

So we have another factor. Now take

$$u(x) = u(x)/g(x) = x^3 + 2x^2 + 3x + 4.$$

At this point, since we know that $a(x)$ has three factors because our basis for W has three elements, we have the complete factorization. However, we can check that the Zassenhaus method works in this case by computing

$$\mathrm{GCD}(a(x), v(x) - 4) = x^3 + 2x^2 + 3x + 4.$$

So each of the three elements of S gives one of the factors of

$$a(x) = (x + 1)(x^2 + 5x + 3)(x^3 + 2x^2 + 3x + 4).\square$$

The above method reduces the number of GCD calculations to only those which are necessary. However, generating the minimal polynomial for $v(x)$ requires substantial computation. Therefore, in order for this algorithm to be useful in practice we need that the probability that we will find a $v(x) \in$ W such that the minimal polynomial of $v(x)$ has degree k is relatively high. Based on experimental data, we think that for a given $a(x)$, the number of $v(x) \in$ W whose minimal polynomials are of degree k is $q \cdot (q-1) \cdots (q-k+1)$ and therefore the probability that this is the case for a random $v(x)$ is

$$(\prod_{i=0}^{k-1}(q - i))/q^k.$$

We leave it as an exercise to show that based on this probability, assuming that q is large, if we want that when $v(x)$ is chosen at random from W that its minimal polynomial has degree k at least half the time, then k must be at most $1.17\sqrt{q}$. However, if k larger in comparison to q, then the probability that a random $v(x) \in$ W has a minimal polynomial of degree k is small. That is, we will have to compute minimal polynomials for many $v(x)$ before finding one that will help us completely factor $a(x)$. Hence if the number of factors of $a(x)$ is small in comparison to q (the number of GCD

computations required when the exhaustive search method is at its worst), then the gain is significant. In the case that k is not small compared to q, we do not know whether there is an efficient method of finding all roots of $a(x)$ using Zassenhaus's minimal polynomial technique.
The proof of the following will be left as an exercise:
Theorem 8.10: Given a monic, square-free polynomial $a(x)$ of degree n, in GF(q)[x], the cost of obtaining the irreducible factorization of $a(x)$ using Zassenhaus's minimal polynomial method, assuming that k is small in comparison to q, is at most $O(n^3 + n^2 \cdot \log(q))$.

The big prime Berlekamp algorithm relies on a method subsequently developed by Cantor and Zassenhaus which determines GCD pairs which will produce nontrivial results with a certain probability, more efficiently. This method, a generalization of Rabin's, is again based on the observation that for q odd

$$x^q - x = x \cdot (x^{(q-1)/2} - 1) \cdot (x^{(q-1)/2} + 1)$$

and the fact that this implies that for any $v(x) \in$ W we have

$$v(x) \cdot (v(x)^{(q-1)/2} - 1) \cdot (v(x)^{(q-1)/2} + 1) = v(x)^q - v(x) \equiv 0 \bmod a(x).$$

We might expect that the nontrivial common factors of $v(x)^q - v(x)$ and $a(x)$ would be spread amongst $v(x), (v(x)^{(q-1)/2} - 1)$ and $(v(x)^{(q-1)/2} + 1)$ in such a way that almost half are factors of the second and almost half are factors of the third, since each of these has degree about half that of $v(x)^q - v(x)$. In fact we have the following:

Theorem 8.11: The probability of GCD$(v(x)^{(q-1)/2} - 1, a(x))$ being nontrivial for $v(x) \in$ W is

$$1 - (\frac{q-1}{2q})^k - (\frac{q+1}{2q})^k.$$

In particular the probability is at least 4/9.

Proof: Let

$$(s_1, \ldots, s_k) = (v(x) \bmod a_1(x), \ldots, v(x) \bmod a_k(x)),$$

where the $a_i(x)$ are the irreducible factors of $a(x)$. Since each $a_i(x)$ is irreducible, each $V_i = GF(q)[x]/ < a_i(x) >$ is a field and hence $s_i \in W_i = \{s \in V_i : s^q = s\}$ must be constant, i.e. $s_i \in GF(q)$. Let

$$w(x) = \text{GCD}(v(x)^{(q-1)/2} - 1, a(x)).$$

Suppose $a_i(x)$ is a factor of $w(x)$. Then

$$a_i(x) \mid (v(x)^{(q-1)/2} - 1)$$

which implies

$$v(x)^{(q-1)/2} - 1 \equiv 0 \bmod a_i(x).$$

Hence

$$v(x)^{(q-1)/2} \equiv 1 \bmod a_i(x)$$

and so $s_i^{(q-1)/2} = 1$. That is, s_i is a quadratic residue of GF(q). Therefore if all s_i for $i = 1 \ldots k$ are quadratic residues, then all $a_i(x)$ divide $w(x)$ and hence $w(x) = a(x)$ and is trivial. Similarly if none of the s_i are quadratic residues, then none of the $a_i(x)$ divide $w(x)$ and hence $w(x) = 1$ and is also trivial. In any other case, at least one and less than k of the factors divide the GCD, and hence the GCD is nontrivial. For each W_i, $(q-1)/2$ of the q elements are quadratic residues and the remaining $(q+1)/2$ are non-quadratic residues. So the probability that an element is a quadratic residue is $(q-1)/2q$ and is a non-quadratic residue is $(q+1)/2q$. When all elements of W_i are equally likely for each component of $(s_1, \ldots, s_k)$, then the probability that all the s_i are quadratic residues is $(\frac{q-1}{2q})^k$ and that all are non-quadratic residues is $(\frac{q+1}{2q})^k$. Therefore the probability that neither of these occurs, that is the probability that $w(x)$ is nontrivial, is

$$1 - (\frac{q-1}{2q})^k - (\frac{q+1}{2q})^k.$$

From the binomial expansion we get that this is equivalent to

$$\begin{aligned}
&1 - (\frac{1}{2q})^k \{(q-1)^k + (q+1)^k\} \\
&= 1 - (\frac{1}{2q})^k \{2q^k + 2\tbinom{k}{2} \cdot q^{k-2} + 2\tbinom{k}{4} \cdot q^{k-4} + \ldots + 2\} \\
&= 1 - \frac{1}{2^{k-1}} \{1 + \tbinom{k}{2} \cdot q^{-2} + \tbinom{k}{4} \cdot q^{-4} + \ldots + q^{-k}\} \\
&\geq 1 - \frac{1}{2}\{1 + \frac{1}{9}\} = \frac{4}{9}
\end{aligned}$$

where the last inequality holds because $k \geq 2$ and $q \geq 3$. □

Note that in the statement of Theorem 8.11 we could replace $\mathrm{GCD}(v(x)^{(q-1)/2} - 1, a(x))$ by $\mathrm{GCD}(v(x)^{(q-1)/2} + 1, a(x))$ and the proof would be equally valid.

To see that no better bound can be found, we give an example for which the probability is exactly $\frac{4}{9}$. Suppose

$$a(x) = x^2 - 1 = (x-1)(x+1) \in \mathrm{GF}(3)[x].$$

So we have $k = 2$ and $q = 3$. Using the Berlekamp method we can also determine that $v^{[1]} = 1,\ v^{[2]} = x$ is a basis for W. That means that the nine elements of W are $\{0, 1, -1, x, -x, x+1, x-1, -x+1, -x-1\}$. For $v(x) = 1$, we get that $\mathrm{GCD}(v(x)^{(q-1)/2} - 1 = v - 1 = 1 - 1 = 0, a(x)) = a(x)$, which is trivial. Then if $v(x)$ is one of $-1, 0, x+1$ or $-x+1$, we get a trivial GCD of one. The other four of the total of nine possibilities for $v(x)$ give non-trivial results.

$$\mathrm{GCD}(x-1, x^2-1) = x-1, \quad \mathrm{GCD}(-x-1, x^2-1) = x+1$$
$$\mathrm{GCD}((x-1)-1, x^2-1) = x+1, \quad \mathrm{GCD}((-x-1)-1, x^2-1) = x-1.$$

For q even, that is $q = 2^m$ for some positive integer m, the above method does not apply since $(q-1)/2$ is not an integer. However, we can factor $x^q - x$ into two factors, each of which has degree approximately $q/2$. To see how we first need to define the trace polynomial over $\mathrm{GF}(2^m)$ by

$$Tr(x) = x + x^2 + x^4 + \ldots + x^{2^{m-1}}.$$

Lemma 8.3: The trace polynomial $Tr(x)$ defined on $\mathrm{GF}(2^m)$ satisfies:
(a) For any v and w in $\mathrm{GF}(2^m)$, $Tr(v+w) = Tr(v) + Tr(w)$;
(b) For any v in $\mathrm{GF}(2^m)$, $Tr(v) \in \mathrm{GF}(2)$;
(c) $x^{2^m} - x = Tr(x) \cdot (Tr(x) + 1)$.

Proof:
(a)

$$\begin{aligned} Tr(v+w) &= v + w + (v+w)^2 + (v+w)^4 + \ldots + (v+w)^{2^{m-1}} \\ &= v + w + v^2 + w + 2 + v^4 + w^4 + \ldots v^{2^{m-1}} + w^{2^{m-1}} \\ &= v + v^2 + v^4 + \ldots + v^{2^{m-1}} + w + w^2 + w^4 + \ldots + w^{2^{m-1}} \\ &= Tr(v) + Tr(w) \end{aligned}$$

(b)

$$\begin{aligned} (Tr(v))^2 &= (v + v^2 + \ldots v^{2^{m-1}})^2 \\ &= v^2 + v^4 + \ldots v^{2^m} \\ &= v^2 + v^4 + \ldots + v^{2^{m-1}} + v \\ &= Tr(v) \end{aligned}$$

So $Tr(v)$ is a solution to $x^2 - x = 0$. We know that in GF(2^m), the only solutions to this equation are in the set of constants $\{0, 1\}$ = GF(2). This implies that for any $v \in$ GF(2^m), $Tr(v)$ is equal to one of the constants in GF(2).
(c) Let $\alpha \in$ GF(2^m). From (b) we have that $Tr(\alpha) = 0$ or 1. Therefore α is a root of the polynomial $Tr(x) \cdot (Tr(x) + 1)$ for all α. So

$$x^{2^m} - x = \prod_{\alpha \in \mathrm{GF}(2^m)} (x - \alpha) \text{ divides } Tr(x) \cdot (Tr(x) + 1).$$

Since $Tr(x)$ and $Tr(x)+1$ are both polynomials of degree 2^{m-1}, their product is of degree 2^m, and so we have that $x^{2^m} - x$ divides another polynomial of the same degree, since both are monic, this implies they must be equal. That is

$$Tr(x) \cdot (Tr(x) + 1) = x^{2^m} - x.$$

Now for the case where q is even, we might expect that common factors of $x^q - x$ and $a(x)$ would be split evenly between $Tr(x)$ and $Tr(x) + 1$ since each is about half the size of $x^q - x$, and in fact:

Theorem 8.12: The probability of GCD($Tr(v(x)), a(x)$), for a random $v(x) \in W$, being nontrivial is

$$1 - (\frac{1}{2})^{k-1}.$$

In particular, the probability is at least 1/2.

Proof: Let

$$(s_1, \ldots, s_k) = (v(x) \bmod a_1(x), \ldots, v(x) \bmod a_k(x)),$$

where the $a_i(x)$ are the irreducible factors of $a(x)$. From Lemma 3(a), $Tr(v(x) \bmod a_i(x)) = Tr(v(x)) \bmod a_i(x)$. So

$$(Tr(v(x)) \bmod a_1(x), \ldots, Tr(v(x)) \bmod a_k(x)) = (Tr(s_1), \ldots, Tr(s_k)).$$

Now from Lemma 3(b), each $Tr(s_i)$ is either 0 or 1.

$$\begin{aligned} \text{And } Tr(s_i) = 0 &\Leftrightarrow Tr(v(x) \bmod a_i(x)) = 0 \\ &\Leftrightarrow Tr(v(x)) \bmod a_i(x) = 0 \\ &\Leftrightarrow a_i(x) \mid Tr(v(x)). \end{aligned}$$

From this we see that if $Tr(s_i) = 0$ then $a_i(x) \mid Tr(v(x))$, for all $i = 1 \ldots k$, and hence $a(x) \mid Tr(v(x))$ so GCD$(Tr(v(x)), a(x))= a(x)$ is trivial. Similarly, if $Tr(s_i) = 1$ for all i then none of the $a_i(x)$ divides $Tr(v(x))$ and the GCD$(Tr(v(x)), a(x))= 1$ is again trivial. The probability of either of these occurrences is $(\frac{1}{2})^k$ and in any other case the GCD is nontrivial, so the overall probability of a nontrivial GCD is

$$1 - 2 \cdot (\frac{1}{2})^k = 1 - (\frac{1}{2})^{k-1},$$

which is at least $1/2$ for any positive integer value of k. $\square$

So the big prime Berlekamp factoring algorithm works by first calculating the Q matrix using binary powering, then finding a basis $\{\mathbf{v}^{[1]}, \ldots, \mathbf{v}^{[k]}\}$ for the null space of $Q - I$, as before. Then we choose a random $v(x) \in$ W by generating random constants $c_i \in$ GF(q) and let

$$v(x) = c_1 \cdot v^{[1]}(x) + c_2 \cdot v^{[2]}(x) + \ldots c_k \cdot v^{[k]}(x).$$

Then calculate

$$w(x) = \text{GCD}(v(x)^{(q-1)/2} - 1, a(x))$$

if q is odd and $w(x) =$ GCD$(Tr(v(x)), a(x))$ if q is even. We know that $w(x)$ will be nontrivial a minimum of about half of the time, so if it is trivial we just pick a new $v(x)$. Once we have a nontrivial $w(x)$ then decompose $a(x)$ into $w(x)$ and $a(x)/w(x)$ and continue using this method to decompose those factors of $a(x)$ until we have determined the k irreducible factors.

Algorithm 8.8: Big Prime Berlekamp Factoring Algorithm.

procedure BigPrimeBerlekamp$(a(x), p^m)$
#Given a monic, square-free polynomial $a(x) \in$ GF(p^m)[x]
#calculate irreducible factors $a_1(x), \ldots, a_k(x)$ such
#that $a(x) = a_1(x) \cdots a_k(x)$ using the big prime
#variation of Berlekamp's algorithm.
Calculate the matrix Q using binary powering
Compute $\{\mathbf{v}^{[1]}, \mathbf{v}^{[2]}, \ldots, \mathbf{v}^{[k]}\}$ a basis for the null space of $Q - I$
#Note: we can ensure that $\mathbf{v}^{[1]} = (1, 0, \ldots, 0)$.
$factors \leftarrow \{a(x)\}$
while SizeOf$(factors) < k$ **do** {
foreach $u(x) \in factors$ **do** {
Choose $\{c_1, \ldots, c_k\}$ at random from (GF(p^m))
$v(x) \leftarrow c_1 v^{[1]}(x) + c_2 v^{[2]}(x) + \ldots + c_k v^{[k]}(x)$

if $p = 2$ **then**
$v(x) \leftarrow v(x) + v(x)^2 + \ldots + v(x)^{2^{m-1}}$
else $v(x) \leftarrow v(x)^{(p^m-1)/2} - 1 \bmod u(x)$
$g(x) \leftarrow \mathrm{GCD}(v(x), u(x))$
if $g(x) \neq 1$ **and** $g(x) \neq u(x)$ **then** {
$w(x) \leftarrow u(x)/g(x)$
$factors \leftarrow factors - \{u(x)\} \cup \{g(x), w(x)\}$
if SizeOf($factors$)$= k$ **then return**($factors$) }}}
end

Example 8.8: To demonstrate this algorithm for q odd, we will again use the same example as in Examples 8.6 and 8.7:

$$a(x) = x^6 - 3x^5 + x^4 - 3x^3 - x^2 - 3x + 1 \in \mathrm{GF}(11)[x].$$

We have already determined the Q matrix and the basis $\{\mathbf{v}^{[1]}, \mathbf{v}^{[2]}, \mathbf{v}^{[3]}\}$. Consider the random element in W given by

$$v(x) = 2v^{[1]}(x) - v^{[2]}(x) - 5v^{[3]}(x) = -5x^5 - x^4 - 2x^3 - 3x^2 - x + 2.$$

Then calculate $\mathrm{GCD}(a(x), v(x)^5 - 1) = 1$. Since this GCD is trivial we choose a new random linear combination

$$v(x) = 1v^{[1]}(x) + 4v^{[2]}(x) - 2v^{[3]}(x) = -2x^5 + 4x^4 - 3x^3 + x^2 + 4x + 1.$$

Then this time the

$$g(x) = \mathrm{GCD}(a(x), v(x)^5 - 1) = x^3 + 2x^2 + 3x + 4.$$

So we have a found a factor of $a(x)$ and after dividing out by this factor

$$w(x) = a(x)/(x^3 + 2x^2 + 3x + 4) = x^3 - 5x^2 - 3x + 3,$$

we have decomposed the degree six polynomial into two degree three factors. At this point we know that there are three irreducible factors, since the basis has three elements, but we don't know which of $w(x)$ or $g(x)$ needs to be factored further, so we continue by trying to factor each of the two with different random $v(x) \in \mathrm{W}$. Trying

$$v(x) = 2v^{[1]}(x) + 3v^{[2]}(x) + 4v^{[3]}(x) = 4x^5 + 3x^4 - 5x^3 - 2x^2 + 3x + 2$$

and $\mathrm{GCD}(g(x), v(x)^5 - 1) = 1$ so we have no new information. Then try $v(x) = 3v^{[1]} + 2v^{[2]} - 2v^{[3]}$. From this we get that $\mathrm{GCD}(w(x), v(x)^5 - 1) =$

$x+1$. Now take $u(x)/(x+1) = x^2+5x+3$ and we have that the irreducible factorization of $a(x)$ into three factors is

$$a(x) = (x+1) \cdot (x^2+5x+3) \cdot (x^3+2x^2+3x+4). \square$$

Example 8.9: We will present a small example to demonstrate the algorithm for an even value of q. Take GF(4) to be the field $\mathbf{Z}_2[x]/ < x^2+x+1 >$ and α to be a root of x^2+x+1. Then the field consists of the elements $\{0, 1, \alpha, 1+\alpha\}$. Let

$$a(x) = x^3 + \alpha x + (\alpha+1) \in \mathrm{GF}(4)[x].$$

To factor this polynomial using Algorithm 8.7 we first need to compute the Q matrix. To do this we use binary powering to compute x^4 mod $a(x)$ and x^8 mod $a(x)$ and we get that

$$Q = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \alpha+1 & \alpha \\ 0 & \alpha & \alpha+1 \end{pmatrix}.$$

Then computing the null space of $Q-I$ we find that $v^{[1]}(x) = 1$ and $v^{[2]}(x) = x + x^2$ form a basis for Q. Now choose c_1 and c_2 at random from GF(4) to form

$$v(x) = c_1 v^{[1]}(x) + c_2 v^{[2]}(x) = (1+\alpha) \cdot 1 + (\alpha) \cdot (x^2+x) = 1 + \alpha + \alpha x + \alpha x^2.$$

Since $q = 4 = 2^2$ in this case $m = 2$ and $Tr(v(x)) = v(x)+v(x)^2$. Calculating $\mathrm{GCD}(a(x), Tr(v(x)))= 1$, we get a trivial GCD, and so we need to choose a new $v(x)$ at random. This time try

$$v(x) = 1 \cdot 1 + (\alpha+1) \cdot (x+x^2) = 1 + (\alpha+1)x + (\alpha+1)x^2.$$

In this case we get that $\mathrm{GCD}(a(x), Tr(v(x)))= x+1$ and we have found one of the factors. To find the other, we simply divide $a(x)/(x+1) = x^2+x+\alpha+1$, and we have the irreducible factorization:

$$a(x) = (x+1) \cdot (x^2+x+\alpha+1). \square$$

Theorem 8.13: The big prime Berlekamp algorithm for factoring a polynomial $a(x)$ of degree n, in the domain GF(q), with k irreducible factors is $\mathrm{O}(n^2 \cdot \log(q) \cdot \log(k) + n^3)$.

Proof: We have seen that determining the Q matrix using binary powering costs $O(n^2 \cdot \log(q) + n^3)$ operations and that determining a basis for W is equivalent to finding the solution space of an $n \times n$ matrix which requires another $O(n^3)$ operations. The cost of computing each random $v(x)$ is linear. Then we start by factoring $a(x)$ into two lower degree factors, each a product of approximately half of the irreducible factors of $a(x)$. Then we do the same to each of the two new factors, repeatedly until we have k irreducible factors. At each step, if q is odd, we calculate $(v(x)^{(q-1)/2} - 1)$ modulo some polynomials (the factors we have determined up to this point) of degree less than or equal to n. At most one of the factors is of degree $O(n)$ at each step so the cost of each step is $O(n^2 \cdot \log(q))$. If q is even, then we compute $v(x) + v(x)^2 + \ldots + v(x)^{2^{m-1}}$ and take that modulo some factor with degree at most n. This is again $O(n^2 \cdot \log(q))$ since we are raising $v(x)$ to $2^{m-1} = q/2 \in O(q)$. Then, for any q, we take the greatest common divisor of the updated $v(x)$ and the same polynomial factor we previously divided out by, which is an additional $O(n^2)$ operations. Since at each step each factor is divided approximately in half, there, on average, a total of $O(\log(k))$ steps. So we have the total cost is
$O(n^2 \cdot \log(q) + n^3) + O(n^3) + O((n^2 \cdot \log(q) + n^2) \cdot \log(k))$

$$\in O(n^2 \cdot \log(q) \cdot \log(k) + n^3). \square$$

8.6 Distinct Degree Factorization

The final method is distinct degree factorization. Given that we have a monic, square-free polynomial $a(x) \in \mathrm{GF}(q)[x]$, the first step of this method is to obtain a partial factorization

$$a(x) = \prod a_i(x)$$

where each $a_i(x)$ is the product of all the irreducible factors of $a(x)$ which have degree i. The second step takes each of the $a_i(x)$ and factors it into irreducible components. To describe the technique used for the first step we first need:

Theorem 8.14: The polynomial $p_r(x) = x^{q^r} - x$ is the product of all monic, irreducible polynomials f in $\mathrm{GF}(q)[x]$ such that the degree of f divides r.

Proof: To prove Theorem 8.14, we will first show that $x^{q^r} - x$ is square-free. Then we will prove that a monic, irreducible polynomial of degree d

divides $p_r(x)$ if and only if $d \mid r$. Let $\tilde{q} = q^r$. Then by Fermat's little theorem (Lemma 8.1) we have that

$$p_r(x) = x^{\tilde{q}} - x = \prod_{a \in \mathrm{GF}(\tilde{q})} (x - a).$$

Now suppose there exists some $g \in \mathrm{GF}(q)[x]$ such that $g^2 \mid p_r(x)$, then $g \mid p_r(x)$. This implies that $(x - a) \mid g$ for some $a \in \mathrm{GF}(\tilde{q})$ and therefore $(x - a)^2 \mid g^2 \mid p_r(x)$ but this is contradiction since $x^{q^r} - x$ is square-free in $\mathrm{GF}(q^r)[x]$, so $x^{q^r} - x$ is square-free in $\mathrm{GF}(q)[x]$.
Now it remains to show that a monic, irreducible polynomial f of degree d divides $p_r(x)$ if and only if d divides r.
First suppose that $f \in \mathrm{GF}(q)[x]$ is a monic, irreducible polynomial of degree d such that $f \mid p_r(x)$. Then since

$$p_r(x) = \prod_{a \in \mathrm{GF}(\tilde{q})} (x - a),$$

it must be that

$$f = \prod_{\hat{a} \in A} (x - \hat{a})$$

where A is some subset of $\mathrm{GF}(\tilde{q})$. Now we choose some root $\hat{a} \in A$ of f and consider the subfield $G = \mathrm{GF}(q)(\hat{a})$ of $\mathrm{GF}(\tilde{q})$. Then $\mathrm{GF}(\tilde{q}) = \mathrm{GF}(q^r)$ is an extension field of G, which has q^d elements since $G \cong \mathrm{GF}(q)[x]/ < f >$ and f had degree d. This implies that the number of elements of $\mathrm{GF}(\tilde{q})$, which is q^r, must be equal to $(q^d)^e$ for some integer e. Therefore $r = de$ for some integer e, or equivalently $d \mid r$.
Next suppose that $d \mid r$ and $f \in \mathrm{GF}(q)[x]$ is a monic, irreducible polynomial of degree d. Since f has degree d,

$$\mathrm{F} = \mathrm{GF}(q)[x]/ < f > \cong \mathrm{GF}(q^d)$$

is a finite field of degree q^d. Let $a \in \mathrm{F}$, then by Fermat's little theorem (Lemma 8.1) $a^{q^d} = a$. Also since $d \mid r$ we know that $r = d \cdot e$ for some integer e. This means that

$$a^{q^r} = ((a^{q^d})^{q^d} \cdots)^{q^d} = a.$$

Consider, in particular, $a = [x]$ the representative element of x in F. Then we know that $[x]^{q^r} = [x]$, which implies $[x^{q^r} - x] = [0]$. Hence $x^{q^r} - x \equiv 0 \bmod f$. And therefore $f \mid p_r(x) = x^{q^r} - x$. $\square$

Theorem 8.14 provides an obvious method for the partial factorization step of the distinct degree method. The algorithm determines $a_i(x)$ by taking increasing integer values of i starting with $i = 1$ and replacing $a(x)$ by $a(x)/a_i(x)$ after each computation of $\mathrm{GCD}(a(x), x^{q^i} - x) = a_i(x)$. It is then only necessary to find the greatest common divisor up to $i = \mathrm{degree}(a(x)/2)$ since at that point, the remaining polynomial must either be trivial or irreducible, since it has no irreducible factors of any smaller degree. The GCD calculations are done by first reducing $x^{q^i} - x$ modulo $a(x)$ for each i. A natural way to reduce x^{q^i} modulo $a(x)$ is by taking the q-th power of $x^{q^{i-1}} \bmod a(x)$, that is

$$(x^{q^i} - x) \bmod a(x) \equiv (x^{q^{i-1}} \bmod a(x))^q - x \bmod a(x).$$

Algorithm 8.9: Distinct Degree Factorization (Part 1: Partial Factorization)

procedure PartialFactorDD($a(x), q$)
  #Given a square-free polynomial $a(x)$ in GF(q)[x],
  #we calculate the partial distinct degree factorization
  #$a_1(x) \cdots a_d(x)$ of $a(x)$
  $i \leftarrow 1$; $w(x) \leftarrow x$; $a_0(x) \leftarrow 1$
  while $i \leq \mathrm{degree}(a(x))/2$ **do**
    $w(x) \leftarrow w(x)^q \bmod a(x)$
    $a_i(x) \leftarrow \mathrm{GCD}(a(x), w(x) - x)$
    if $a_i(x) \neq 1$ **then**
      $a(x) \leftarrow a(x)/a_i(x)$
      $w(x) \leftarrow w(x) \bmod a(x)$
    $i = i + 1$
  return$(a_0(x) \cdots a_{i-1}(x) a(x))$
end

Example 8.10: Let $a(x) = x^{27} - 1 \in \mathrm{GF}(7)[x]$. Then

$$a_1(x) = \mathrm{GCD}(a(x), x^7 - 1) = x^3 - 1$$

which tells us that $a(x)$ has three linear factors. Then replace $a(x)$ by

$$a(x)/a_1(x) = x^{24} + x^{21} + x^{18} + x^{15} + x^{12} + x^9 + x^6 + x^3 + 1.$$

Next we find that the polynomial has no quadratic factors, since

$$\mathrm{GCD}(a(x), x^{49} - x) = 1,$$

and two cubic factors, since

$$\mathrm{GCD}(a(x), x^{243} - x) = x^6 + x^3 + 1.$$

Let

$$a(x) = a(x)/a_3(x) = x^{18} + x^9 + 1.$$

From here we find that $\mathrm{GCD}(a(x), x^{2401} - x) = 1$, $\mathrm{GCD}(a(x), x^{16807} - x) = 1$, $\mathrm{GCD}(a(x), x^{117649} - x) = 1$, $\mathrm{GCD}(a(x), x^{823543} - x) = 1$ and $\mathrm{GCD}(a(x), x^{5764801} - x) = 1$, so there are no irreducible factors of degree 4,5,6,7, or 8. And finally

$$a_9(x) = \mathrm{GCD}(a(x), x^{40353607} - x) = x^{18} + x^9 + 1$$

which means that the remaining part of $a(x)$ is the product of two irreducible factors of degree nine and we have completed the partial factorization of $a(x)$.

Theorem 8.15: The complexity of the partial factorization step of the distinct degree algorithm for factoring a polynomial of degree n into factors $a_i(x)$ where each $a_i(x)$ is the product of all irreducible factors of $a(x)$ which have degree i, over $\mathrm{GF}(q)$, is at most $\mathrm{O}(n^3 \cdot \log(q))$.

Proof: Each time through the loop, to get x^{q^i} for the new i, we raise the value, $(x^{q^{i-1}})$, to the power of q modulo $a(x)$. Since $a(x)$ has degree n the first time through the loop and at most n after that, the cost of this operation is $\mathrm{O}(n^2 \cdot \log(q))$ and the cost of both the GCD calculation and the division of $a(x)$ by $a_i(x)$ is $\mathrm{O}(n^2)$. If $a(x)$ is irreducible or the product of two factors, each of which has degree $n/2$, then it will require $n/2$ times through the loop to gain any information. This is the maximum since in any other case the degree of $a(x)$ will be reduced before i reaches $n/2$. This means that the total cost is at most

$$\mathrm{O}(n^2 \cdot \log(q) + n^2)\frac{n}{2} \in \mathrm{O}(n^3 \cdot \log(q)).$$

Note that if the degree of the largest factor is l, then the cost is $\mathrm{O}(n^2 \cdot \log(q) \cdot l)$. □

For q large, there is a better method, as pointed out by Lenstra. Given the Q matrix, as defined previously, it can easily be shown that

$$\mathbf{v} \cdot Q \equiv \mathbf{v}^q \bmod a(x)$$

for any polynomial $v(x) \in \mathrm{GF}(q)/ < a(x) >$, where $\mathbf{v}$ and $\mathbf{v}^q$ are the coefficient vectors of the representatives of $v(x)$ and $v(x)^q$, respectively, of smallest degree in the residue ring V. Using this we can reduce the computation of raising $x^{q^{i-1}}$ to the q modulo $a(x)$ to that of multiplying the vector $\mathbf{v}$ by the matrix Q. This matrix multiplication costs $\mathrm{O}(n^2)$ operations rather than the $\mathrm{O}(n^2 \cdot \log(q))$ of the other method. Now, since using this modification reduces all the steps within the loop to a cost of $\mathrm{O}(n^2)$, the complexity of the modified algorithm is dominated by the cost of generating the Q matrix, which we know to be $\mathrm{O}(n^2 \cdot \log(q) + n^3)$. Comparing this to the complexity $\mathrm{O}(n^3 \cdot \log(q))$ of the original algorithm, we see that the modification is an improvement in cases when q is very large relative to n.

From the first step of distinct degree factorization we have a method for separating $a(x)$ into factors $a_i(x)$, each of which is a product of irreducibles of degree i. To complete the factorization of $a(x)$, we need a way to factor the $a_i(x)$ which are not already irreducible into irreducible polynomials, each of degree i. To do this we can again use the method of Cantor and Zassenhaus, with the trace polynomial $Tr(x)$ as defined in the section on the big prime Berlekamp algorithm for q even. Suppose q is odd and $a_i(x)$ is reducible, that is that its degree is at least $2i$. We can generalize one direction of the proof of Theorem 8.14 to show that for any polynomial $v(x)$, any irreducible polynomial of degree d, where $d \mid i$, divides $v(x)^{q^i} - v(x)$. From this we see that $v(x)^{q^i} - v(x)$ is a multiple of all irreducible polynomials of degree i, in particular, every irreducible factor of $a_i(x)$ divides $v(x)^{q^i} - v(x)$. Therefore, since we can factor $v(x)^{q^i} - v(x)$ as $v(x) \cdot (v(x)^{(q-1)/2} - 1) \cdot (v(x)^{(q-1)/2} + 1)$, we can factor $a_i(x)$ as

$$\mathrm{GCD}(a_i(x), v(x)) \cdot \mathrm{GCD}(a_i(x), v(x)^{(q-1)/2} - 1) \cdot \mathrm{GCD}(a_i(x), v(x)^{(q-1)/2} + 1).$$

As with the big prime Berlekamp algorithm, if we choose $v(x)$ with degree at most $2 \cdot i - 1$, then $\mathrm{GCD}(a_i(x), v(x)^{(q-1)/2} - 1)$ is nontrivial, and hence we get a nontrivial factor of $a_i(x)$, approximately half the time. Now suppose q is even, that is $q = 2^m$ and again the degree of $a_i(x)$ is at least $2i$. From Lemma 3, we have that over $\mathrm{GF}(2^{m \cdot i})$, where $Tr(x) = x + x^2 + \ldots + x^{2^{mi-1}}$,

$$x^{q^i} - x = x^{2^{mi}} - x = Tr(x) \cdot (Tr(x) + 1).$$

So, if $v(x)$ is any polynomial of degree at most $2 \cdot i - 1$, then we have

$$a_i(x) = \mathrm{GCD}(a_i(x), Tr(v(x))) \cdot \mathrm{GCD}(a_i(x), Tr(v(x)) + 1).$$

And by calculating $\mathrm{GCD}(a_i(x), Tr(x))$ for a random $v(x)$ with this property, we get a nontrivial factor of $a_i(x)$ with probability at least $1/2$.

Using the above, we get the following:

Algorithm 8.10: Distinct Degree Factorization (Part II: Splitting Factors.)

procedure SplitDD($a_i(x), i, p^m$)
  # Given a polynomial $a_i(x) \in \mathrm{GF}(p^m)[x]$ which is
  # made up of irreducible factors of degree i, we split $a_i(x)$
  # into its complete factorization via Cantor-Zassenhaus method
  if $\deg(a_i(x), x) \leq i$ **then return**($\{a_i(x)\}$)
  $factors \leftarrow \{a(x)\}$
  repeat {
    Choose $v(x) \in \mathrm{GF}(q)[x]$ of degree $2i-1$ at random
    if $p = 2$ **then**
      $v(x) \leftarrow v(x) + v(x)^2 + \ldots v(x)^{2^{mi-1}}$
    else
      $v(x) \leftarrow v(x)^{(q^i-1)/2} - 1$
    $g(x) \leftarrow \mathrm{GCD}(a_i(x), v(x))$
    if $g(x) \neq 1$ **and** $g(x) \neq a_i(x)$ **then** {
      $factors \leftarrow \mathrm{SplitDD}(g(x), i, p^m) \cup \mathrm{SplitDD}(a_i(x)/g(x), i, p^m)\}\}$
  return($factors$)
end

Applying Algorithm 8.10 to each reducible $a_i(x)$ determined by Algorithm 8.9 gives a complete algorithm for factoring a polynomial $a(x)$ into its irreducible components.

Example 8.11: We will demonstrate this method by completing the factorization of the polynomial from Example 8.10. So we have

$$a(x) = a_1(x) \cdot a_3(x) \cdot a_9(x) = (x^3 - 1) \cdot (x^6 + x^3 + 1) \cdot (x^{18} + x^9 + 1).$$

First we choose a random $v(x)$ of degree at most 1. Try $v(x) = x + 3$, then compute

$$\mathrm{GCD}(x^3 - 1, (x+3)^3 - 1) = x - 1.$$

So $x - 1$ is one of the linear factors, leaving $(x^3 - 1)/(x - 1) = x^2 + x + 1$. Now take $v(x) = x + 1$ and

$$\mathrm{GCD}(x^2 + x + 1, (x+1)^3 - 1) = 1$$

which is trivial so we need to choose a new $v(x)$, say $v(x) = x + 2$. Then

$$\mathrm{GCD}(x^2 + x + 1, (x + 2)^3 - 1) = x - 2$$

so the remaining two linear factors of $a_1(x)$, and therefore of $a(x)$, are $x - 2$ and $(x^2 + x + 1)/(x - 2) = x + 3$. That is

$$a_1(x) = (x - 1)(x - 2)(x + 3).$$

Next we know that $a_3(x)$ has degree 6 and therefore $a(x)$ has two irreducible cubic factors. To split $a_3(x)$ into those two factors, first try $v(x) = x - 3$ and calculate

$$\mathrm{GCD}(x^6 + x^3 + 1, (x - 3)^{(7^3-1)/2} - 1) = 1$$

so we need to choose another $v(x)$. If $v(x) = x - 2$ then

$$\mathrm{GCD}(x^6 + x^3 + 1, (x - 2)^{(7^3-1)/2} - 1) = x^3 - 2$$

and so we have that the cubic factors are $x^3 - 2$ and $a_3(x)/(x^3 - 2) = x^3 + 3$. Similarly we can also determine that $a_9(x) = (x^9 - 2)(x^9 + 3)$ and hence that

$$a(x) = (x - 1)(x - 2)(x + 3)(x^3 - 2)(x^3 + 3)(x^9 - 2)(x^9 + 3)$$

is the complete factorization of $a(x) = x^{27} - 1$ over GF(7). □

Theorem 8.16: The complexity of the splitting algorithm for factoring a polynomial $a_i(x)$, which is the product of $k > 1$ monic, irreducible factors, each of degree i, over a field GF(q) is $\mathrm{O}(i^3 \cdot k^2 \cdot \log(q))$.

Proof: The cost of producing a random polynomial $v(x)$ is linear. For both q odd and q even, updating the $v(x)$ polynomial in the following step involves raising $v(x)$ to the power of q^i, modulo $a_i(x)$, and so costs $\mathrm{O}((i \cdot k)^2 \cdot \log(q^i) = i^3k^2\log(q))$, since $a_i(x)$ has degree $i \cdot k$. Then the GCD calculation requires $\mathrm{O}((i \cdot k)^2)$ operations since $i \cdot k$ is the degree of $a_i(x)$ and $v(x)$ is taken modulo $a_i(x)$. This is repeated with two different $v(x)$ on average to get nontrivial GCD, and once we have a nontrivial factor, $a_i(x)$ is split. Then, as with the big prime Berlekamp algorithm, we need to continue to apply the algorithm to each factor after the split, repeatedly until we have found all k irreducible factors. Since at each level, the polynomial is split approximately in half, the second time we need to split two degree $(i \cdot k)/2$ polynomials. So the cost of this step is $\mathrm{O}(2 \cdot (\frac{ik}{2})^2\log(q))$,

which is half the cost of the first step. Similarly at the next step, we split four factors, each with a quarter of the original degree, so this step costs a quarter of the first. So the complexity is dominated by the cost of the first step, as the sum of the costs of all subsequent steps is approximately equal to the cost of the first step alone. That is, the total cost is $O(i^3 \cdot k^2 \cdot \log(q))$. □

Suppose $a(x)$ is a monic, square-free polynomial of degree n. To better understand the cost of the algorithm when we combine the two parts to get a complete factorization into irreducible components we consider four cases and give their complexities, without proof, in a table:

$a(x)$ composed of	Complexity	
	Partial Factorization	Splitting
$n/2$ quadratic factors	$O(n^2 \log(q))$	$O(n^2 \log(q))$
$\sqrt{n}$ factors of degree $\sqrt{n}$	$O(n^{5/2} \log(q))$	$O(n^{5/2} \log(q))$
2 factors of degree $n/2$	$O(n^3 \log(q))$	$O(n^3 \log(q))$
one irreducible factor	$O(n^3 \log(q))$	-

References

[1] K.O. Geddes, S.R. Czapor and G. Labahn, *Algorithms for Computer Algebra*, Kluwer Academic Publishers, Norwell, MA, USA, 1992.

[2] J. von zur Gathen and J. Gerhard, *Modern Computer Algebra*, Cambridge Unviersity Press, Cambridge, 2003.

[3] Michael Rabin. Probabilistic Algorithms in Finite Fields. *SIAM J. Computing* **9**(2) 273–280, 1980.