On the deterministic complexity of factoring polynomials[†]

Shuhong Gao

Department of Mathematical Sciences

Clemson University

Clemson, SC 29634 USA

sgao@math.clemson.edu

(March 17, 1999)

The paper focuses on the deterministic complexity of factoring polynomials over finite fields assuming the extended Riemann hypothesis (ERH). By the works of Berlekamp (1967, 1970) and Zassenhaus (1969), the general problem reduces deterministically in polynomial time to finding a proper factor of any squarefree and completely splitting polynomial over a prime field $\mathbb{F}_p$. Algorithms are designed to split such polynomials. It is proved that a proper factor of a polynomial can be found deterministically in polynomial time, under ERH, if its roots do not satisfy some stringent condition, called *super square balanced*. It is conjectured that super square balanced polynomials do not exist.

1. Introduction

We consider the problem of factoring polynomials over finite fields. This problem can be solved in probabilistic polynomial time (Berlekamp 1970, Cantor and Zassenhaus 1981, von zur Gathen and Shoup 1992, Kaltofen and Shoup 1998), but it is still open whether it has a deterministic polynomial time algorithm, even if the extended Riemann hypothesis (ERH) is assumed. We are interested in the deterministic complexity of the problem under ERH.

Various authors have given under ERH efficient algorithms for special classes of polynomials or for polynomials over special fields. Rónyai (1992) showed under ERH that any polynomial with integer coefficients that generates a Galois number field can be factored mod p in deterministic polynomial time, except for finitely many primes p, which extends previous results by Huang (1991), Adleman, Mander & Miller (1977), and Evdokimov (1989). If the number of irreducible factors of a polynomial is bounded, Rónyai (1988) showed under ERH that it can be factored deterministically in polynomial time. On special fields, Bach, von zur Gathen & Lenstra (1995) showed that polynomials over finite fields of characteristic p can be factored in polynomial time if $\Phi_k(p)$ is smooth for some integer k where $\Phi_k(x)$ denotes the k-th cyclotomic polynomial, which extends the works of von zur Gathen (1987), Moenck (1977), Camion (1983), Mignotte & Schnorr (1988),

[†] Part of the work was done while the author was an NSERC Postdoctoral Fellow at the University of Toronto and a Visiting Assistant Professor at the University of Waterloo; their hospitality and support are gratefully acknowledged.

and Rónyai (1989). Recently, Evdokimov (1994) proved that every polynomial over $\mathbb{F}_q$ of degree n can be factored deterministically in time polynomial in $n^{\log n}$ and $\log q$.

In this paper, we continue this line of research for deterministic polynomial time algorithms under ERH. By the algorithms of Berlekamp (1967, 1970) and Zassenhaus (1969), the general problem can be reduced to finding proper factors of polynomials that split completely over prime fields. To be precise, we focus on the following problem. For any given prime p and a polynomial $f \in \mathbb{F}_p[x]$ that is squarefree and splits completely over $\mathbb{F}_p$, find a proper factor of f. Under ERH, Rónyai (1988) proves that, for any completely splitting polynomial $f \in \mathbb{F}_p[x]$ of degree n and any prime divisor $r|n$, a proper factor of f can be found in time polynomial in n^r and $\log p$. In particular, if $r = 2$ this means that any completely splitting polynomial $f \in \mathbb{F}_p[x]$ of even degree can be split in polynomial time under ERH. However, when n has no small divisors, say n is a prime, then Rónyai's time is exponential in n. That is where the current paper contributes. We design algorithms that terminate in polynomial time under ERH. It is proven that if f does not satisfy some stringent conditions then our algorithms will always find a proper factor of f. For simplicity, we state our result here only for the case $p \equiv 3 \bmod 4$. The general statement can be found in Theorem 3.7.

Suppose that $p \equiv 3 \bmod 4$ is a prime. Let F be a subset of $\mathbb{F}_p$ with cardinality $n > 1$. We say that F is *square balanced* if, for each $\xi \in F$,

$$\#\{\zeta \in F : \zeta \neq \xi,\ \xi - \zeta \text{ is a square in } \mathbb{F}_p\} = \frac{n-1}{2}.$$

Two sets $F_1, F_2 \subset \mathbb{F}_q$, each with cardinality at least two, are called *mutually square balanced* if, for each $\xi \in F_1$,

$$\#\{\zeta \in F_2 : \xi - \zeta \text{ is a square in } \mathbb{F}_p\}$$

is the same for all $\xi \in F_1$, and similarly for each $\xi \in F_2$ with $\zeta \in F_1$. For a subset $F \subset \mathbb{F}_p$ and an integer k, define $F_k = \{a^k : a \in F\}$, the set of k-th powers of the elements in F. We call a subset $F \subset \mathbb{F}_p$ of cardinality $n > 1$ *super square balanced* if the following three conditions are satisfied:

(i) For each $1 \leq k \leq (n \log p)^6$, F_k has cardinality n and is square balanced;
(ii) All the sets F_k, $1 \leq k \leq (n \log p)^6$, are pairwise disjoint;
(iii) All the sets F_k, $1 \leq k \leq (n \log p)^6$, are pairwise mutually square balanced.

Note that a squarefree and completely splitting polynomial $f \in \mathbb{F}_p[x]$ factors as $\prod_{i=1}^{n}(x - \xi_i)$ where $\xi_1, \ldots, \xi_n$ are different elements in $\mathbb{F}_p$. Thus squarefree and completely splitting polynomials in $\mathbb{F}_p[x]$ are in 1-1 correspondence to the subsets of $\mathbb{F}_p$. We call a squarefree and completely splitting polynomial $f \in \mathbb{F}_p[x]$ *square balanced* or *super square balanced* if the set of its roots is square balanced or super square balanced, respectively. We prove the following theorem.

THEOREM 1.1. *Given any prime $p \equiv 3 \bmod 4$ and $f \in \mathbb{F}_p[x]$ squarefree and completely splitting, we can find a proper factor of f in deterministic polynomial time provided that ERH holds and f is not super square balanced.*

Rónyai's result for $r = 2$ follows from the above theorem immediately, since if f has even degree then f can not even be square balanced!

Theorem 1.1 puts stringent conditions on the roots of polynomials that can not be split in polynomial time by our algorithms under ERH. An interesting number theory problem arises here, that is, whether there exists any super square balanced set in $\mathbb{F}_p$. We believe that the conditions **(i), (ii)** and **(iii)** are so strong that no subset in $\mathbb{F}_p$ can satisfy them all. We conjecture that for any prime p and any positive integer n, there is no super square balanced subset in $\mathbb{F}_p$ of cardinality $n > 1$. A confirmation to the conjecture implies that, under ERH, polynomials over finite fields can be factored deterministically in polynomial time.

We should point out that our results are purely theoretical. They bear on an issue in theoretical computer science about derandomization, namely, if a problem can be solved efficiently by randomized algorithms, can it also be solved efficiently by algorithms without using randomness? In our case, the goal is to decide whether there is a deterministic polynomial time algorithm to factor polynomials over finite fields. For such a purpose, it is satisfactory when an algorithm runs in polynomial time and thus we do not attempt to implement our algorithms in the most efficient fashion. For practical purposes, it suffices to use the randomized algorithms of Berlekamp (1970), Cantor and Zassenhaus (1981), von zur Gathen and Shoup (1992), Kaltofen and Shoup (1998).

The rest of the paper is organized as follows. In Section 2, we discuss the arithmetic of polynomials over algebras. In the current paper, we work mainly with semisimple algebras over $\mathbb{F}_p$ that split completely. We show how to adapt the gcd concept for polynomials over a field to polynomials over an algebra and modify the Euclidean algorithm to compute gcd. We examine the zero structure of a completely splitting polynomial over an algebra and answer such questions as how many roots it has, how many decompositions it has, and which set of roots form a decomposition. We also define the characteristic polynomial of an element in an algebra over a subalgebra and show a simple formula under an orthogonal basis. These properties are used in Section 3 in algorithm design and analysis. They are also of independent interest and may be useful elsewhere. Interestingly, Wan (1996) uses characteristic polynomials in a different fashion to factor polynomials over finite fields. In Section 2.4, we review a method for computing k-th roots of elements in a semisimple algebra and define the concept of square balanced and super square balanced polynomials over an arbitrary finite field $\mathbb{F}_q$. In Section 3, we describe our algorithms and their analysis, our main results are proved there.

2. Arithmetic of polynomials over algebras

When computing in an algebra $\mathcal{R}$ of dimension n over $\mathbb{F}_p$, by "polynomial time" we mean that the number of $\mathbb{F}_p$-operations used is bounded above by a polynomial in n and $\log p$, i.e., $(n \log p)^{O(1)}$. We also say "efficient" to mean "polynomial time".

We say that an algebra $\mathcal{R}$ is explicitly given if we know a basis of $\mathcal{R}$ and the product of any two basis elements expressed under the same basis. Thus addition and multiplication in $\mathcal{R}$ can be done in polynomial time. Identity element and inverse of an invertible element in $\mathcal{R}$ can also be computed efficiently by solving a system of linear equations over $\mathbb{F}_p$.

When factoring polynomials over $\mathbb{F}_p$, we work in the algebra $\mathcal{R} = \mathbb{F}_p[x]/(f)$ where $f \in \mathbb{F}_p[x]$ is squarefree and completely splitting. In this paper, we also work in extension algebras of $\mathcal{R}$. These algebras are special cases of semisimple commutative algebras. In general, let $\mathbb{F}$ be any field. We call an algebra $\mathcal{R}$ over $\mathbb{F}$ *an elementary algebra* if there are primitive idempotents $\mu_1, \ldots, \mu_n$ such that $\mathcal{R} = \mathbb{F}\mu_1 \oplus \cdots \oplus \mathbb{F}\mu_n$. This means that $\mathcal{R}$ has

a unique basis over $\mathbb{F}$ such that addition and multiplication are computed componentwise under this basis. Note that primitive idempotents of $\mathcal{R}$ are unique.

A monic polynomial $g \in \mathcal{R}[x]$ of degree n is called completely splitting if $g = \prod_{i=1}^{n}(x - C_i)$ for some $C_i \in \mathcal{R}$, and g is called separable if $C_i - C_j$ is not a zero divisor in $\mathcal{R}$ for all $i \neq j$. For any elementary algebra $\mathcal{R}$ and any $g \in \mathcal{R}[x]$ monic, separable and completely splitting, it is easy to prove that $\mathcal{R}[x]/(g)$ is an elementary algebra.

2.1. GCD of polynomials

In this section, we discuss the gcd concept of polynomials over an elementary algebra. At the surface, this does not seem to make any sense, since the polynomial ring over an elementary algebra is not an integral domain, not to mention a unique factorization domain. Due to the presence of zero divisors, a polynomial of degree n over an elementary algebra can be written as a product of polynomials of degrees greater than n. Hence one has to be very careful when dealing with the concept of gcd. It turns out that we can still use the usual definition of gcd and the Euclidean algorithm can be adapted to compute gcd of polynomials over an elementary algebra.

We need some terminology. Let $\mathcal{R}$ be an elementary algebra of dimension n over a field $\mathbb{F}$ with primitive idempotents $\mu_1, \ldots, \mu_n$. For any element $A \in \mathcal{R}$, there exist unique elements $a_1, \ldots, a_n \in \mathbb{F}$ such that

$$A = \sum_{i=1}^{n} a_i \mu_i.$$

We call a_i the ith canonical projection of A into $\mathbb{F}$, denoted by A_i. For any polynomial $f \in \mathcal{R}[x]$, f_i denotes the polynomial in $\mathbb{F}[x]$ with each coefficient being the ith projection of the corresponding coefficient of f. Thus $f = \sum_{i=1}^{n} f_i \mu_i$.

LEMMA 2.1. *Let $f, g \in \mathcal{R}[x]$. Then*

(a) $(f+g)_i = f_i + g_i$, $(fg)_i = f_i g_i$, *for* $1 \leq i \leq n$;
(b) $f|g$ *iff* $f_i|g_i$ *for all* $1 \leq i \leq n$;
(c) $\deg f_i = \deg f$, $1 \leq i \leq n$, *iff the leading coefficient of f is invertible in $\mathcal{R}$.*

Proof. They follow directly from the fact that $\sum_{i=1}^{n} \mu_i = 1$, and that, for any $A, B \in \mathcal{R}$,

$$(A+B)_i = A_i + B_i, \quad (AB)_i = A_i B_i, \quad 1 \leq i \leq n.$$

(Addition and multiplication in $\mathcal{R}$ are computed componentwise.) □

A direct consequence of this lemma is that one can characterize all the zero divisors in $\mathcal{R}[x]$: a polynomial is a zero divisor if and only if at least one of its canonical projections is zero.

We can now define gcd as follows. Let $f, g \in \mathcal{R}[x]$. Any common divisor of f, g that is divisible by every common divisor is called a gcd of f and g. We call a polynomial in $\mathcal{R}[x]$ *pseudo-monic* if each of its canonical projections is either monic or 0.

THEOREM 2.2. *For any $f, g \in \mathcal{R}[x]$, there is a unique pseudo-monic gcd of f, g.*

Proof. We first prove the existence. Let $f_i, g_i \in \mathbb{F}[x]$ be the i-th canonical projections of

f, g, respectively. Let $\gcd(f_i, g_i)$ denote the conventional gcd, thus monic or zero. Then it is easy to check that the polynomial

$$\sum_{i=1}^{n} \gcd(f_i, g_i)\mu_i. \tag{2.1}$$

is a gcd of f, g and is pseudo-monic (here we assume that $\gcd(0, 0) = 0$).

To prove the uniqueness, suppose that h is any gcd of f, g with all canonical projections monic or zero. We prove that the i-th projection h_i of h is equal to $\gcd(f_i, g_i)$, $1 \le i \le n$. As $h|f$ and $h|g$, we have $h_i|f_i$ and $h_i|g_i$ and $h_i|\gcd(f_i, g_i)$. Now let d_i be any common factor of f_i and g_i, and let

$$d = \mu_1 + \cdots \mu_{i-1} + d_i\mu_i + \mu_{i+1} + \cdots + \mu_n.$$

Then $d \in \mathcal{R}[x]$ divides both f and g. Thus $d|h$ and consequently $d_i|h_i$. Therefore $h_i = \gcd(f_i, g_i)$. (If $f_i = g_i = 0$ then we can take any polynomial as d_i, so h_i must be 0.) □

We use $\gcd(f, g)$ to denote the unique pseudo-monic gcd of f and g. By the above proof, $\gcd(f, g)$ is given by (2.1). Hence $\gcd(f, g)$ is monic iff the degree of $\gcd(f_i, g_i)$ is the same for all $1 \le i \le n$.

The next question is how to compute $\gcd(f, g)$ for any given $f, g \in \mathcal{R}[x]$. If the primitive idempotents of $\mathcal{R}$ are known then this is trivial, just using the Euclidean algorithm and the formula (2.1). In practice, we do not know them, and $\mathcal{R}$ is represented by some other basis. This does not present any difficulty at all. We can modify the Euclidean algorithm as follows.

Suppose that $f, g \in \mathcal{R}[x]$ with $\deg f \ge \deg g$ and we want to compute $\gcd(f, g)$. If the leading coefficient of g is invertible in $\mathcal{R}$, then division by g can be carried out as usual without any trouble. Suppose that the leading coefficient, say a, of g is a zero divisor in $\mathcal{R}$. We can first compute the identity elements I_1, I_2 of the two subalgebras:

$$\mathcal{R}_1 = \{ra : r \in \mathcal{R}\}, \quad \mathcal{R}_2 = \{r \in \mathcal{R} : ra = 0\}.$$

Explicit bases for the two subalgebras can be computed by solving linear systems of equations over $\mathbb{F}$. If $\mathcal{R}$ is represented under the basis of primitive idempotents μ_i's and $a = a_1\mu_1 + \cdots + a_n\mu_n$, then $\mathcal{R}_1$ is generated by all the μ_i's where a_i's are not zero, and $\mathcal{R}_2$ is generated by those μ_i's where a_i's are zero. Hence $\mathcal{R}_1$ and $\mathcal{R}_2$ are orthogonal complements in $\mathcal{R}$, that is, $\mathcal{R} = \mathcal{R}_1 \oplus \mathcal{R}_2$ and $r_1r_2 = 0$ for all $r_1 \in \mathcal{R}_1$, $r_2 \in \mathcal{R}_2$. Let

$$f_1 = fI_1, g_1 = gI_1 \in \mathcal{R}_1[x], \quad f_2 = fI_2, g_2 = gI_2 \in \mathcal{R}_2[x].$$

Then

$$\gcd(f, g) = \gcd(f_1, g_1) + \gcd(f_2, g_2).$$

Apply the algorithm recursively in the subalgebras to compute $\gcd(f_1, g_1)$ and $\gcd(f_2, g_2)$. Note that the leading coefficient of g_1 is aI_1 and is invertible in $\mathcal{R}_1$ and the degree of g_2 is smaller than the degree of g, as $aI_2 = 0$. When $\mathcal{R}$ is an elementary algebra over $\mathbb{F}_p$, this modified Euclidean algorithm runs in polynomial time.

2.2. Zero structure of polynomials

Let $\mathcal{R}$ be an elementary algebra of dimension n over a field $\mathbb{F}$ with primitive idempotents $\mu_1, \ldots, \mu_n$. Let $g \in \mathcal{R}[x]$ be completely splitting over $\mathcal{R}$, i.e.,

$$g = \prod_{i=1}^{m} (x - C_i), \quad C_i \in \mathcal{R}. \tag{2.2}$$

We want to know how many zeroes g has in $\mathcal{R}$, which set of zeroes form a decomposition (2.2) and how many different decompositions g has.

As $\mu_1, \ldots, \mu_n$ form a basis for $\mathcal{R}$ over $\mathbb{F}$, there are unique elements $c_{ij} \in \mathbb{F}$ such that

$$C_i = \sum_{j=1}^{n} c_{ij}\mu_j, \quad 1 \le i \le m.$$

For $1 \le j \le n$, define

$$\begin{aligned} C^{[j]} &= \{c_{1j}, \ldots, c_{mj}\}, \\ g_j &= \prod_{i=1}^{m} (x - c_{ij}) \in \mathbb{F}[x]. \end{aligned}$$

Lemma 2.3. *An element $A = \sum_{j=1}^{n} a_j\mu_j$, $a_j \in \mathbb{F}$, is a zero of g iff $a_j \in C^{[j]}$ for $1 \le j \le n$. Hence g has $\prod_{j=1}^{n} |C^{[j]}|$ different zeroes in $\mathcal{R}$.*

Proof. Note that $(x - A) | g$ iff $(x - a_j) | g_j$, $1 \le j \le n$, and the latter is true iff $a_j \in C^{[j]}$, $1 \le j \le n$. □

Lemma 2.4. *For $1 \le i \le m$, let $A_i = \sum_{j=1}^{n} a_{ij}\mu_j \in \mathcal{R}$ where $a_{ij} \in \mathbb{F}$. Then $g = \prod_{i=1}^{m}(x - A_i)$ iff each column of the $m \times n$ matrix (a_{ij}) is a permutation of the corresponding column of (c_{ij}). Therefore g has $\prod_{j=1}^{n} k_j$ different decompositions (2.2) over $\mathcal{R}$ where k_j is the number of different permutations of the jth column of (c_{ij}).*

Proof. Note that

$$\prod_{i=1}^{m}(x - A_i) = \sum_{j=1}^{n} \left(\prod_{i=1}^{m} (x - a_{ij}) \right) \mu_j,$$

and

$$g = \prod_{i=1}^{m}(x - A_i) \text{ iff } \prod_{i=1}^{m}(x - a_{ij}) = g_j, \ 1 \le j \le n.$$

As $\prod_{i=1}^{m}(x - a_{ij})$ and g_j are polynomials over a field, $a_{1j}, \ldots, a_{mj}$ must be a permutation of $c_{1j}, \ldots, c_{mj}$, the roots of g_j. The lemma follows immediately. □

Theorem 2.5. *Let $\mathcal{R}$ be an elementary algebra of dimension n over a field $\mathbb{F}$ with primitive idempotents $\mu_1, \ldots, \mu_n$. Suppose that $g = \prod_{i=1}^{m}(x - c_i) \in \mathbb{F}[x]$ where $c_1, \ldots, c_m$ are different elements in $\mathbb{F}$. Then*

(a) *$A = \sum_{j=1}^{n} a_j\mu_j \in \mathcal{R}$, $a_j \in \mathbb{F}$, is a zero of g iff $a_j \in \{c_1, \ldots, c_m\}$, $1 \le j \le n$;*

(b) *For $1 \le i \le m$, let $A_i = \sum_{j=1}^{n} a_{ij}\mu_j \in \mathcal{R}$ where $a_{ij} \in \mathbb{F}$. Then $g = \prod_{i=1}^{m}(x - A_i)$ iff each column of the $m \times n$ matrix (a_{ij}) is a permutation of $c_1, \ldots, c_m$.*

Proof. It follows from Lemmas 2.3 and 2.4 and the fact that, for any element $a \in \mathbb{F}$, $a = \sum_{j=1}^{n} a\mu_j$ in $\mathcal{R}$. □

2.3. Characteristic polynomials

Let $\mathcal{T}$ be a commutative algebra of dimension n over an elementary algebra $\mathcal{R}$. For any $\alpha \in \mathcal{T}$, the characteristic polynomial of α over $\mathcal{R}$ is defined to be that of the mapping: $\xi \mapsto \alpha\xi$, $\xi \in \mathcal{T}$, which is an $\mathcal{R}$-module homomorphism. The characteristic polynomial of an element is invariant with respect to different bases. It can be computed with any explicitly given basis $(\alpha_1, \ldots, \alpha_n)$ of $\mathcal{T}$ over $\mathcal{R}$ as follows. Compute

$$\alpha \cdot \alpha_i = \sum_{j=1}^{n} m_{ij}\alpha_j, \quad m_{ij} \in \mathcal{R}, \quad 1 \le i \le n.$$

Then $\det(I_n x - (m_{ij}))$ is the characteristic polynomial of α. The determinant can be computed in polynomial time (again, go to subalgebras when necessary).

Characteristic polynomials have a very simple formula under an orthogonal basis. By an orthogonal basis of $\mathcal{T}$ over $\mathcal{R}$, we mean some elements $\mu_1, \ldots, \mu_n$ such that $\mathcal{T} = \mathcal{R}\mu_1 \oplus \cdots \oplus \mathcal{R}\mu_n$ and $\mu_i\mu_j = 0$ for $i \ne j$ and $\mu_i^2 = 1$ for all i. For any $\alpha \in \mathcal{T}$, there are unique elements $a_i \in \mathcal{R}$ such that $\alpha = \sum_{i=1}^{n} a_i\mu_i$. Note that $\alpha \cdot \mu_i = a_i\mu_i$, $1 \le i \le n$. We see that the characteristic polynomial of α is

$$\alpha(x) = \prod_{i=1}^{n}(x - a_i) \in \mathcal{R}[x]. \tag{2.3}$$

$\mathcal{T}$ may have many orthogonal bases over $\mathcal{R}$, but the formula (2.3) is true for any of them. We will use this formula in Section 3 when analyzing our algorithms. As noted above, the polynomial in (2.3) can be computed in polynomial time by using any explicitly given basis, without knowing an orthogonal basis.

2.4. Finding roots and square balanced polynomials

Let $\mathcal{R}$ be an elementary algebra of dimension n over a finite field $\mathbb{F}_q$ where q is a prime power. We need to efficiently compute k-th roots of elements in $\mathcal{R}$ for various integers k. Evdokimov (1994) shows that this can be done under ERH in $(nk \log q)^{O(1)}$ operations in $\mathbb{F}_q$, where ERH is used only to construct an r-th power nonresidue in $\mathbb{F}_q$ for every prime divisor r of $\gcd(k, q-1)$. Evdokimov's algorithm is a direct generalization of Adleman, Manders and Miller (1977) and Pohlig and Hellman (1978).

We describe a slightly modified version of Evdokimov's algorithm here so that we can observe some of its properties. These properties will be useful later in analyzing the algorithms in Section 3. It suffices to show how to find an r-th root of an arbitrary element in $\mathcal{R}$ for any prime r. If r is coprime to $q-1$ then A^s is an r-th root of A where $sr \equiv 1 \bmod q-1$. So we assume henceforth that r is a prime and $r|(q-1)$. Suppose that $q - 1 = r^e w$ where $r \nmid w$. Let η be a fixed primitive r^e-th root of unity in $\mathbb{F}_q$. We remark that η can be taken as ξ^w for any primitive root or r-th nonresidue ξ in $\mathbb{F}_q$ and ξ can be constructed efficiently assuming ERH (Wang 1959, Bach 1997).

Note that an element $a \in \mathbb{F}_q$ has an r-th root in $\mathbb{F}_q$ iff $a = 0$ or $a^{(q-1)/r} = 1$. When $a \neq 0$, write a as

$$a = \eta^u \theta$$

for some integer u with $0 \leq u < r^e$ and $\theta \in \mathbb{F}_q$ with $\theta^w = 1$. Then $a^{(q-1)/r} = 1$ iff $r|u$. To see what happens in $\mathcal{R}$, let $\mu_1, \ldots, \mu_n$ be the primitive idempotents of $\mathcal{R}$ over $\mathbb{F}_q$ and $A = \sum_{i=1}^n a_i \mu_i$ where $a_i \in \mathbb{F}_q$. For any $B = \sum_{i=1}^n b_i \mu_i$ where $b_i \in \mathbb{F}_q$, we have $b^r = \sum_{i=1}^n b_i^r \mu_i$. So $B^r = A$ iff $b_i^r = a_i$ for $1 \leq i \leq n$. The latter is true iff $a_i^{(q-1)/r} = 0$ or 1 for $1 \leq i \leq n$, i.e., $A^{(q-1)/r}$ is an idempotent in $\mathcal{R}$ (each component is 0 or 1).

Now we show how to find roots of A. If $a_i = 0$ for some i then certainly $b_i = 0$. So we only need to work with the nonzero components of A. Consider the subalgebra $\mathcal{R}A = \{CA : C \in \mathcal{R}\}$. Let I be the identity element of $\mathcal{R}A$. Then $A \cdot I$ is invertible in $\mathcal{R}A$ and an r-th root of $A \cdot I$ in $\mathcal{R}A$ is an r-th root of A in $\mathcal{R}$.

Henceforth we assume that $A \in \mathcal{R}$ is invertible. We assume that an r-th root of A exists in $\mathcal{R}$, which means that $A^{(q-1)/r}$ is the identity element 1 in $\mathcal{R}$. Find integers s and t such that $sr^e + tw = 1$. Then

$$A = A^{tw}(A^{sr^{e-1}})^r.$$

It suffices to find an r-th root of A^{tw}. Denote $\bar{A} = A^{tw}$. Note that $\bar{A}^{r^e} = (A^{q-1})^t = 1$. Find the smallest integer $k \geq 0$ such that $\bar{A}^{r^k} \in \mathbb{F}_q$. Then $\bar{A}^{r^k}$ is a power of η. Use Pohlig and Hellman's algorithm to find an integer u such that

$$\bar{A}^{r^k} = \eta^u.$$

Since A has an r-th root in $\mathcal{R}$, u must be divisible by r. If $k = 0$ then $\eta^{u/r}$ is an r-th root of $\bar{A}$. If $k > 0$, then find a zero divisor in $\mathcal{R}$ as follows. Let $B = \bar{A}^{r^{k-1}}$ and $\zeta = \eta^{r^{e-1}}$. Then $B \notin \mathbb{F}_q$ and ζ is a primitive r-th root of unity. Note that

$$B\eta^{-u/r} \notin \mathbb{F}_q \quad \text{and} \quad (B\eta^{-u/r})^r = 1.$$

We have $r + 1$ distinct r-th roots of unity in $\mathcal{R}$, i.e., $1, \zeta, \ldots, \zeta^{r-1}$ and $B\eta^{-u/r}$. So $B\eta^{-u/r} - \zeta^i$ is a zero divisor in $\mathcal{R}$ for some $0 \leq i < r$. We find this i by an exhaustive search. Let $D = B\eta^{-u/r} - \zeta^i$ and

$$\mathcal{R}_1 = \mathcal{R}D = \{DC : C \in \mathcal{R}\}, \quad \mathcal{R}_2 = \{C \in \mathcal{R} : DC = 0\}.$$

Then $\mathcal{R}_1$ and $\mathcal{R}_2$ are nontrivial subalgebras of $\mathcal{R}$ and $\mathcal{R} = \mathcal{R}_1 \oplus \mathcal{R}_2$. Explicit bases for $\mathcal{R}_1$ and $\mathcal{R}_2$ can be computed by solving systems of linear equations. We next compute $\bar{A} = A_1 + A_2$ where $A_1 \in \mathcal{R}_1$ and $A_2 \in \mathcal{R}_2$, and proceed recursively in $\mathcal{R}_1$ and $\mathcal{R}_2$, respectively, to compute r-th roots of A_1 and A_2. The whole process can be finished in time polynomial in r, n and $\log q$.

Let σ_r denote the above algorithm for computing r-th roots by using $\eta \in \mathbb{F}_q$ as a primitive r^e-th root of unity. Denote by $\sigma_r(A) \in \mathcal{R}$ the output of σ_r on input $A \in \mathcal{R}$. Then $(\sigma_r(A))^r = A$ if A has an r-th root in $\mathcal{R}$. When $q \equiv 3 \bmod 4$, η has only one choice, namely, $\eta = -1$. In this case, σ_2 is nothing but the formula: $\sigma_2(A) = A^{(q+1)/4}$ provided that A has a quadratic root in $\mathcal{R}$. In general, observe that the only operations in σ_r on A are powering and canonical projections into subalgebras. We see that σ_r acts individually to each component under the primitive idempotent basis over $\mathbb{F}_q$.

LEMMA 2.6. *Given a primitive r^e-th root η of unity in $\mathbb{F}_q$ where $q - 1 = r^e w$, $e \geq 1$ and*

$r \nmid w$, the algorithm σ_r runs in polynomial time in $r, \log q$ and $n = \dim \mathcal{R}$. Furthermore, σ_r has the following properties:

(a) *$\sigma_r(aA) = \sigma_r(a)A$ for $a \in \mathbb{F}_q$, if $A \in \mathcal{R}$ is idempotent, i.e., $A^2 = A$.*

(b) *$\sigma_r(A + B) = \sigma_r(A) + \sigma_r(B)$, if $A, B \in \mathcal{R}$ are orthogonal, i.e., $AB = 0$.*

(c) *Let $\mu_1, \ldots, \mu_n$ be primitive idempotents in $\mathcal{R}$ and $A = \sum_{i=1}^{n} a_i \mu_i \in \mathcal{R}$ where $a_i \in \mathbb{F}_q$. Then*

$$\sigma_r(A) = \sum_{i=1}^{n} \sigma_r(a_i)\mu_i.$$

(d) *Let $a = \eta^u \theta$ where $\theta \in \mathbb{F}_q$ with $\theta^w = 1$ and $0 \le u < r^e$. Then $\sigma_r(a^r) = a$ iff $u < r^{e-1}$.*

(e) *Suppose q is odd and $a \in \mathbb{F}_q \setminus \{0\}$. Then $\sigma_2(a^2) = \pm a$.*

Proof. Properties (a) and (b) follow from the fact that σ_r acts individually to each primitive component of $\mathcal{R}$ over $\mathbb{F}_q$. (c) follows from (a) and (b). To see (d), write $u = u_0 r^{e-1} + u_1$ where $0 \le u_1 < r^{e-1}$ and $0 \le u_0 \le r - 1$. As

$$a^r = \eta^{u_0 r^e + r u_1} \theta^r = \eta^{r u_1} \theta^r,$$

we see that $\sigma_r(a^r) = \eta^{u_1}\theta$, which is equal to a iff $u = u_1$, i.e., $u < r^{e-1}$. This proves (d). When $r = 2$, we have $\eta^{2^{e-1}} = -1$, so $\eta^{u_0 2^{e-1}} = (-1)^{u_0} = \pm 1$ depending on $u_0 = 0$ or 1. Hence $\sigma_2(a^2) = \eta^{u_1}\theta = \pm a$, which is part (e). □

When $q \equiv 3 \bmod 4$, $\eta = -1$ and the property (e) reads as: $\sigma_2(a^2) = a$ for $a \in \mathbb{F}_q$ iff a is a square in $\mathbb{F}_q$, and $\sigma_2(a^2) = -a$ iff a is not a square in $\mathbb{F}_q$.

This leads to the concept of square balanced and mutually square balanced sets for general q mentioned in the introduction. Let σ_2 be the above deterministic algorithm for computing quadratic roots using a primitive 2^e-th root η of unity in $\mathbb{F}_q$ where 2^e divides $q - 1$ exactly. A subset $F \subset \mathbb{F}_q$ of cardinality $n > 1$ is called *square balanced with respect to η* if, for each $\xi \in F$,

$$\#\left\{\zeta \in F : \zeta \ne \xi,\ \sigma_2\big((\xi - \zeta)^2\big) = \xi - \zeta\right\} = \frac{n-1}{2}.$$

Two sets $F_1, F_2 \subset \mathbb{F}_q$, each with cardinality at least two, are called *mutually square balanced with respect to η* if for each $\xi \in F_1$,

$$\#\left\{\zeta \in F_2 : \sigma_2\big((\xi - \zeta)^2\big) = \xi - \zeta\right\}$$

is the same for all $\xi \in F_1$, and similarly for $\xi \in F_2$ and $\zeta \in F_1$.

When $q \equiv 3 \bmod 4$, this definition agrees with the one given in the introduction. When $q \equiv 1 \bmod 4$, however, $\sigma_2\big((\xi - \zeta)^2\big) = \xi - \zeta$ does not imply that $\xi - \zeta$ is a square in $\mathbb{F}_q$. Also, there are many choices for η and it is possible that a subset of $\mathbb{F}_q$ is square balanced with respect to one choice but not to another. For example, $q = 17$ and $F = \{1, 4, 5\}$. Then F is square balanced with respect to $\eta = 3$ but not to $\eta = 6$. We often omit the reference to η when it is fixed or clear from the context.

As squarefree and completely splitting polynomials in $\mathbb{F}_q[x]$ are in 1-1 correspondence to subsets of $\mathbb{F}_q$, we also say that a squarefree and completely splitting polynomial is *square balanced* if the set of its roots is square balanced. We construct an infinite family of square balanced polynomials.

LEMMA 2.7. *Let $n > 1$ be an odd factor of $q-1$. Then the polynomial $f = x^n - 1$ is always square balanced (with respect to any η).*

Proof. Let ξ be a primitive n-th root of unity in $\mathbb{F}_q$. Then

$$f = x^n - 1 = \prod_{i=0}^{n-1}(x - \xi^i).$$

Let $q - 1 = 2^e w$ where w is odd and η a 2^e-th primitive root of unity in $\mathbb{F}_q$. Define

$$D = \left\{0 \le j \le n-1 : j \ne 0, \sigma_2\big((1-\xi^j)^2\big) = 1 - \xi^j\right\}.$$

For any $i \ne j$, suppose that $1 - \xi^{j-i} = \eta^u \theta$ where $\theta \in \mathbb{F}_q$ with $\theta^w = 1$. As the order n of ξ is an odd factor of $q-1$, we have $n|w$ and so $\xi^w = 1$. Hence

$$\xi^i - \xi^j = \xi^i(1 - \xi^{j-i}) = \eta^u(\xi^i\theta)$$

with $(\xi^i\theta)^w = 1$. By Lemma 2.6 (d),

$$\sigma_2\big((1-\xi^{j-i})^2\big) = 1 - \xi^{j-i} \text{ iff } \sigma_2\big((\xi^i - \xi^j)^2\big) = \xi^i - \xi^j,$$

as each of which holds iff $u < 2^{e-1}$. For $0 \le i \le n-1$,

$$\begin{aligned} D_i &= \left\{0 \le j \le n-1 : j \ne i, \sigma_2\big((\xi^i - \xi^j)^2\big) = \xi^i - \xi^j\right\} \\ &= \left\{0 \le j \le n-1 : j \ne i, \sigma_2\big((1-\xi^{j-i})^2\big) = 1 - \xi^{j-i}\right\} \\ &= \{j : j - i \in D\} = D + i. \end{aligned}$$

Therefore $|D| = |D_1| = \cdots = |D_{n-1}|$. Let $t = |D|$. Note that, for $i \ne j$, $\sigma_2\big((\xi^i - \xi^j)^2\big) = \xi^i - \xi^j$ iff $\sigma_2\big((\xi^j - \xi^i)^2\big) = -(\xi^j - \xi^i)$, so $j \in D_i$ iff $i \notin D_j$. By counting the pairs $j \in D_i$, we have $nt = n(n-1)/2$ and thus $t = (n-1)/2$. This proves the theorem. □

Finally, let $c > 1$ be a constant and F a subset of $\mathbb{F}_q$ with cardinality $n > 1$. We say that F is *c-super square balanced* if the following three conditions are satisfied:

(i) For each $1 \le k \le (n \log q)^c$, F_k has cardinality n and is square balanced;
(ii) The sets F_k, $1 \le k \le (n \log q)^c$, are pairwise disjoint;
(iii) The sets F_k, $1 \le k \le (n \log q)^c$, are pairwise mutually square balanced.

A squarefree and completely splitting polynomial in $\mathbb{F}_q[x]$ is called *c-super square balanced* if its set of roots in $\mathbb{F}_q$ is c-super square balanced. The polynomial $x^n - 1$ above is square balanced but not 2-super square balanced, as (ii) is violated for $k = 1$ and 2.

3. Algorithms and Analysis

Suppose that we want to factor $f \in \mathbb{F}_p[x]$ of degree n where f has n different roots in $\mathbb{F}_p$, say

$$f = \prod_{i=1}^{n}(x - \xi_i), \quad \xi_i \in \mathbb{F}_p.$$

Let

$$\mathcal{R} = \mathbb{F}_p[x]/(f) = \mathbb{F}_p[A]$$

where $A = x \bmod f$. Then $1, A, \ldots, A^{n-1}$ form an explicit basis for $\mathcal{R}$ over $\mathbb{F}_p$. Define

$$f^*(y) = f(y)/(y - A) \in \mathcal{R}[y] \text{ and } \mathcal{T} = \mathcal{R}[y]/(f^*) = \mathcal{R}[B]$$

where $B = y \bmod f^*$. Then $1, B, \ldots, B^{n-2}$ form an explicit basis for $\mathcal{T}$ over $\mathcal{R}$, and $A^i B^j$, $0 \le i \le n-1$, $0 \le j \le n-2$, form an explicit basis for $\mathcal{T}$ over $\mathbb{F}_p$.

Let η be a fixed 2^e-th primitive root of unity in $\mathbb{F}_p$ where 2^e divides $p-1$ exactly, and let σ be the deterministic algorithm σ_2 from Section 2.4 for computing quadratic roots in $\mathcal{T}$. That is, if $C \in \mathcal{T}$ is a square then $\sigma(C)$ is the output of the algorithm which satisfies $(\sigma(C))^2 = C$. The main idea of our algorithms is to employ the property of σ as stated in Lemma 2.6 (c). This property says that, when applied to an element $C \in \mathcal{T}$, σ acts individually to the coordinates of C under the primitive idempotent basis of $\mathcal{T}$ over $\mathbb{F}_p$, and so $\sigma\big((A-B)^2\big) \neq \pm(A-B)$ in general. Such a case usually enables one to find a zero divisor in $\mathcal{R}$ and thus a proper factor of f, via the characteristic polynomial and gcd techniques discussed in Section 2. The construction of C in Step 1 below was motivated by Evdokimov (1994).

Algorithm 3.1

Input: $f \in \mathbb{F}_p[x]$ squarefree and completely splitting over $\mathbb{F}_p$ where p is an odd prime,
Output: a proper factor of f or "Failure".

0. Form A, B, $\mathcal{R}, \mathcal{T}$ as described above.
1. Compute $C = \frac{1}{2}\big(A + B + \sigma((A-B)^2)\big) \in \mathcal{T}$.
2. Compute the characteristic polynomial $c(z)$ of C over $\mathcal{R}$.
3. Decompose $c(z)$ as $c(z) = h(z)(z-A)^t$ where t is the largest possible.
   Set $H = h(A) \in \mathcal{R}$. Then $H \neq 0$.
4. If H is a zero divisor in $\mathcal{R}$ then find a proper factor of f,
   otherwise output "Failure".

THEOREM 3.1. *Algorithm 3.1 terminates in polynomial time under ERH, and outputs "Failure" if and only if f is square balanced.*

Proof. Consider the running time first. By Lemma 2.6, σ finds a quadratic root of $(A-B)^2$ in polynomial time provided that η is given. But η can be constructed efficiently under ERH. So Step 1 can be done in polynomial time under ERH. Steps 2 and 3 can also be finished in polynomial time. For Step 4, one just views $H \in \mathcal{R}$ as a polynomial in $\mathbb{F}_p[x]$ and computes $\gcd(H, f)$. Note that $\gcd(H, f)$ is a proper factor of f if and only if H is a zero divisor in $\mathcal{R}$. So this step can be done in polynomial time too. Hence the whole algorithm runs in polynomial time under ERH. (ERH was used only to construct η.)

To prove the other statement, define

$$t_i = \#\left\{1 \le j \le n : j \neq i, \sigma\big((\xi_i - \xi_j)^2\big) = \xi_i - \xi_j\right\}, 1 \le i \le n. \tag{3.1}$$

We prove that the H in step 3 is not a zero divisor in $\mathcal{R}$ if and only if $t_1 = \cdots = t_n$, and this common value of t_i must equal $(n-1)/2$.

For this purpose, we examine the element $C \in \mathcal{T}$ obtained in Step 1 and characterize the set of zeroes of the polynomial $c(z)$ in $\mathcal{R}$. Order the primitive idempotents $\mu_1, \ldots, \mu_n$ of $\mathcal{R}$ such that

$$A = \sum_{i=1}^{n} \xi_i \mu_i.$$

For $1 \le j \le n-1$, let

$$B_j = \sum_{i=1}^{n} b_{ji}\mu_i \in \mathcal{R},$$

where $b_{ji} \in \{\xi_1, \ldots, \xi_n\}$ such that $(\xi_i, b_{1i}, \ldots, b_{n-1\,i})$ is a permutation of $(\xi_1, \xi_2, \ldots, \xi_n)$ for each $1 \le i \le n$. Then, by Theorem 2.5,

$$f(y) = (y - A)\prod_{j=1}^{n-1}(y - B_j), \quad f^*(y) = \prod_{j=1}^{n-1}(y - B_j).$$

Define

$$\nu_j = \prod_{k\neq j}(B - B_k)/\prod_{k\neq j}(B_j - B_k), \quad 1 \le j \le n-1.$$

Then

$$B = \sum_{j=1}^{n-1} B_j\nu_j,$$

and

$$\sum_{i=1}^{n-1}\nu_i = 1, \quad \nu_i\nu_j = \begin{cases} \nu_i, & \text{if } i = j, \\ 0, & \text{otherwise.} \end{cases}$$

Thus $\nu_1, \ldots, \nu_{n-1}$ form an orthogonal basis for $\mathcal{T}$ over $\mathcal{R}$. Note that

$$A \pm B = \sum_{j=1}^{n-1}(A \pm B_j)\nu_j,$$

and

$$C = \sum_{j=1}^{n-1}\frac{1}{2}\left((A + B_j) + \sigma\big((A - B_j)^2\big)\right)\nu_j.$$

By (2.3), the characteristic polynomial of C over $\mathcal{R}$ is

$$\begin{aligned}
c(z) &= \prod_{j=1}^{n-1}\left(z - \frac{1}{2}\left(A + B_j + \sigma\big((A - B_j)^2\big)\right)\right) \\
&= \prod_{j=1}^{n-1}\left(z - \frac{1}{2}\left(\sum_{i=1}^{n}(\xi_i + b_{ji})\mu_i + \sigma\Big(\sum_{i=1}^{n}(\xi_i - b_{ji})^2\mu_i\Big)\right)\right) \\
&= \prod_{j=1}^{n-1}\sum_{i=1}^{n}\Big(z - \frac{1}{2}\Big(\xi_i + b_{ji} + \sigma\big((\xi_i - b_{ji})^2\big)\Big)\Big)\mu_i \\
&= \sum_{i=1}^{n}\prod_{j=1}^{n-1}\Big(z - \frac{1}{2}\Big(\xi_i + b_{ji} + \sigma\big((\xi_i - b_{ji})^2\big)\Big)\Big)\mu_i \\
&= \sum_{i=1}^{n}\prod_{j\neq i}\Big(z - \frac{1}{2}\Big(\xi_i + \xi_j + \sigma\big((\xi_i - \xi_j)^2\big)\Big)\Big)\mu_i,
\end{aligned}$$

since $(\xi_i, b_{1\,i}, \ldots, b_{n-1\,i})$ is a permutation of $(\xi_1, \xi_2, \ldots, \xi_n)$. Note that

$$\frac{1}{2}\Big(\xi_i + \xi_j + \sigma\big((\xi_i - \xi_j)^2\big)\Big) = \begin{cases} \xi_i, & \text{if } \sigma\big((\xi_i - \xi_j)^2\big) = \xi_i - \xi_j, \\ \xi_j, & \text{if } \sigma\big((\xi_i - \xi_j)^2\big) = -(\xi_i - \xi_j). \end{cases}$$

For $1 \le i \le n$, define

$$\begin{aligned} \Delta_i &= \big\{1 \le j \le n : j \ne i, \sigma\big((\xi_i - \xi_j)^2\big) = -(\xi_i - \xi_j)\big\}, \\ \bar{\Delta}_i &= \big\{1 \le j \le n : j \ne i, \sigma\big((\xi_i - \xi_j)^2\big) = \xi_i - \xi_j\big\}. \end{aligned}$$

Then $t_i = \#\bar{\Delta}_i$ and

$$c(z) = \sum_{i=1}^{n} (z - \xi_i)^{t_i} \prod_{j \in \Delta_i} (z - \xi_j)\mu_i. \tag{3.2}$$

Since

$$z - A = \sum_{i=1}^{n} (z - \xi_i)\mu_i,$$

we see from Lemma 2.1 (b) that the e in step 3 is equal to $\min\{t_1, \ldots, t_n\}$ and

$$h(z) = \sum_{i=1}^{n} (z - \xi_i)^{t_i - t} \prod_{j \in \Delta_i} (z - \xi_j)\mu_i, \quad H = h(A) = \sum_{i=1}^{n} (\xi_i - \xi_i)^{t_i - t} \prod_{j \in \Delta_i} (\xi_i - \xi_j)\mu_i.$$

If $t_i > t$ for some i then the coefficient of μ_i is zero, so H is a zero divisor in $\mathcal{R}$. Obviously, H is not a zero divisor in $\mathcal{R}$ if and only if $t_i = t$ for $1 \le i \le n$.

It remains to prove that if $t_1 = \cdots = t_n = t$ then $t = (n-1)/2$. Note that

$$\sigma\big((\xi_i - \xi_j)^2\big) = \xi_i - \xi_j \;\text{ iff }\; \sigma\big((\xi_j - \xi_i)^2\big) = -(\xi_j - \xi_i) \;\text{ for all } i \ne j.$$

Each pair (i, j), $1 \le i < j \le n$, contributes to either t_i or t_j but not both. This implies that

$$nt = \sum_{i=1}^{n} t_i = n(n-1)/2.$$

Therefore $t = (n-1)/2$. □

COROLLARY 3.2. (RÓNYAI 1992) *If the degree n of f is even, then Algorithm 3.1 finds a proper factor of f.*

When the t_i's are not equal, we can refine Algorithm 3.1 to get more proper factors of f. The polynomial $c(z)$ from Step 2 contains enough information to separate ξ_i from ξ_j whenever $t_i \ne t_j$. More precisely, we have the following algorithm and theorem.

Algorithm 3.2

Input: $f \in \mathbb{F}_p[x]$ squarefree and completely splitting over $\mathbb{F}_p$.

Output: a list of factors of f.

0. Form A, B, $\mathcal{R}, \mathcal{T}$ as described above.
1. Compute $C = \frac{1}{2}\big(A + B + \sigma((A - B)^2)\big) \in \mathcal{T}$.
2. Compute the characteristic polynomial $c(z)$ of C over $\mathcal{R}$.
3. Compute $d(z) = \gcd(c(z), (z - A)^n) \in \mathcal{R}[z]$.
   Suppose that $d(z) = \sum_{k=0}^{n} D_k z^k$, $D_k \in \mathcal{R}$.

Set $d_n = f$.
For k from $n-1$ down to 0 do
Compute $d_k = \gcd(D_k, d_{k+1})$ (D_k is viewed as a polynomial in $\mathbb{F}_p[x]$).
Set $f_k = d_{k+1}/d_k$.
Return $f_0, f_1, \ldots, f_{n-1}$.

THEOREM 3.3. *Under ERH, Algorithm 3.2 factors f as $f = f_0 f_1 \cdots f_{n-1}$ in polynomial time with*

$$f_k(x) = \prod_{1 \le i \le n, t_i = k} (x - \xi_i) \in \mathbb{F}_p[x], 0 \le k \le n-1, \tag{3.3}$$

where the t_i's are defined as in (3.1) and an empty product is assumed to be 1.

Proof. Obviously Step 3 can be finished in polynomial time. By the proof of Theorem 3.1, Steps 1 and 2 can be done in polynomial time under ERH. Therefore Algorithm 3.2 runs in polynomial time under ERH.

We prove that the f_k computed by Algorithm 3.2 is the same as in (3.3). Use the notations in the proof of Theorem 3.1. Since

$$(z - A)^n = \sum_{i=1}^{n} (z - \xi_i)^n \mu_i,$$

it follows from (2.1) and (3.2) that

$$d(z) = \sum_{k=0}^{n} D_k z^k = \gcd\big(c(z), (z-A)^n\big) = \sum_{i=1}^{n} (z - \xi_i)^{t_i} \mu_i.$$

Write

$$D_k = \sum_{i=1}^{n} d_{ik} \mu_i, \quad d_{ik} \in \mathbb{F}_p.$$

Since the i-th canonical projection of $d(z)$ has degree t_i, we see that

$$d_{ik} = 0, \quad \text{if } k > t_i.$$

For any $D = \sum_{i=1}^{n} d_i \mu_i \in \mathcal{R}$, $d_i \in \mathbb{F}_p$, $\gcd(D, f) = \prod_{d_i = 0} (z - \xi_i)$. By induction on k (from $n-1$ down to 0), we have

$$\begin{aligned} d_{k+1} &= \prod_{1 \le i \le n, t_i \le k} (z - \xi_i), \\ d_k &= \gcd(D_k, d_{k+1}) = \prod_{1 \le i \le n, t_i < k} (z - \xi_i). \end{aligned}$$

Therefore

$$f_k = d_{k+1}/d_k = \prod_{1 \le i \le n, t_i = k} (z - \xi_i).$$

This completes the proof. □

Next we apply Algorithms 3.1 and 3.2 to many polynomials related to f to obtain a much stronger result. For any integer $k > 0$, define $f^{(k)} \in \mathbb{F}_p[x]$ to be the polynomial whose roots are the k-th powers of the roots of f. The idea is to apply Algorithm 3.1 to

split $f^{(k)}$. If a proper factor of $f^{(k)}$ is found then use it to get a proper factor of f. We show that this can be done in polynomial time when k is small. We can apply this for many, even all values of k such that $k \leq (n \log p)^{O(1)}$, and the total running time is still polynomial. We also apply Algorithm 3.2 to polynomials $f^{(k)} \cdot f^{(\ell)}$ for $0 < k, \ell \leq (n \log p)^{O(1)}$. Since $f^{(k)} \cdot f^{(\ell)}$ has even degree $2n$, Algorithm 3.2 will output some proper factors of $f^{(k)} \cdot f^{(\ell)}$. If for some pair of k and ℓ, there is one factor not equal to $f^{(k)}$ nor $f^{(\ell)}$, then we can compute a proper factor of $f^{(k)}$ or $f^{(\ell)}$, thus a proper factor of f. To materialize this scheme, we need to show how to compute $f^{(k)}$ efficiently and how to get a proper factor of f when given a proper factor of $f^{(k)}$.

First note that $f^{(k)}(x) = \mathrm{res}_y(f(y), x - y^k)$ where res_y denotes the resultant of polynomials with respect to the variable y. So $f^{(k)}$ can be computed in time polynomial in k, n and $\log p$.

LEMMA 3.4. *Given a proper factor of $f^{(k)}$, we can find a proper factor of f in time polynomial in k, n and $\log p$ assuming ERH.*

Proof. Let h be a given proper factor of $f^{(k)}$. Without loss of generality, we may assume that h is squarefree and

$$h = \prod_{i=1}^{\ell} (x - \xi_i^k).$$

Let $K \subset \mathbb{F}_p$ be the subset

$$K = \{\beta \in \mathbb{F}_p : \beta^k = 1\}.$$

The set K can be computed in time polynomial in k and $\log p$ by factoring the polynomial $x^k - 1$ under ERH (Huang 1991). Form $\mathcal{T} = \mathbb{F}_p[x]/(h)$, and $A = x \bmod h$. Compute a k-th root H of A in $\mathcal{T}$ by the algorithm in Section 2.4. Let g be the characteristic polynomial of H in $\mathcal{T}$ over $\mathbb{F}_p$. Then there exist $a_i \in K$ such that

$$g = \prod_{i=1}^{\ell} (x - a_i \xi_i).$$

For each $a \in K$, compute

$$f_0 = \gcd\big(f(x), g(ax)\big).$$

If $a = a_i$ then $(x - \xi_i) | f_0$. Since g has degree $\ell < n$, this f_0 is a proper factor of f. Under ERH, all these can be done in time $(kn \log p)^{O(1)}$. □

LEMMA 3.5. *Suppose $f \in \mathbb{F}_p[x]$ is squarefree and completely splitting. Let $\mathcal{R} = \mathbb{F}_p[x]/(f)$ and $A = x \bmod f$. For an integer k, if $A^k \neq A$ and the characteristic polynomial of A^k over $\mathbb{F}_p$ is equal to f then $A \mapsto A^k$ induces a nontrivial endomorphism of $\mathcal{R}$.*

Proof. Since f is the minimal polynomial of A over $\mathbb{F}_p$, we just need to check whether $f(A^k) = 0$, but it is true as f is equal to the characteristic polynomial of A^k over $\mathbb{F}_p$ by our assumption. □

We also need the following result:

LEMMA 3.6. (RÓNYAI 1992) *Given any prime p and a polynomial $f \in \mathbb{F}_p[x]$ square-*

free and completely splitting, together with a nontrivial endomorphism of the algebra $\mathbb{F}_p[x]/(f)$, a proper factor of f can be found in polynomial time under ERH.

We are now ready to describe our next algorithm.

Algorithm 3.3

Input: $f \in \mathbb{F}_p[x]$ of degree n, squarefree and completely splitting over $\mathbb{F}_p$,
and a constant $c > 1$.
Output: a proper factor of f or "Failure".

0. Form $\mathcal{R} = \mathbb{F}_p[A]$ where $A = x \bmod f$.
Apply Algorithm 3.1 to f. If a proper factor of f is found then halt.
1. For each integer $1 < k \le (n \log p)^c$, compute $A^k \in \mathcal{R}$.
1.1. If $A^k = A$ then f is a factor of $x^{k-1} - 1$; factor f (by factoring $x^{k-1} - 1$) and halt. Otherwise, compute the characteristic polynomial $f^{(k)}$ of A^k over $\mathbb{F}_p$.
1.2. If $f^{(k)}$ is not squarefree, find a proper factor of $f^{(k)}$ and then
Find a proper factor of f and halt.
1.3. Compute $d = \gcd(f, f^{(k)})$. If $d \neq 1$ or f then output d and halt.
If $d = f$ then $f = f^{(k)}$ and, by Lemma 3.5, $A \mapsto A^k$ induces
a nontrivial endomorphism of $\mathcal{R}$; find a proper factor of f and halt.
1.4. Apply Algorithm 3.1 to the polynomial $f^{(k)}$. If it outputs a proper factor of $f^{(k)}$ then find a proper factor of f and halt.
2. For each integer pair $0 < k < \ell \le (n \log p)^c$ do the following:
2.1. Compute $d = \gcd(f^{(k)}, f^{(\ell)})$.
If d is a proper factor then use it to find a proper factor of f and halt.
If $d = f^{(k)} = f^{(\ell)}$ then either $A^k = A^\ell$ (so the roots of f are $(\ell - k)$-th roots of unity in $\mathbb{F}_p$) or $A^k \mapsto A^\ell$ induces a nontrivial automorphism of $\mathcal{R}$;
in both cases, find a proper factor of f and halt.
2.2. Apply Algorithm 3.2 to $g = f^{(k)} \cdot f^{(\ell)} \in \mathbb{F}_p[x]$ to get a list of factors of g.
2.3. For each factor h in the list, compute $u = \gcd(h, f^{(k)})$ and $v = \gcd(h, f^{(\ell)})$.
If u or v is a proper factor of $f^{(k)}$ or $f^{(\ell)}$, respectively,
then use it to find a proper factor of f and halt.

Otherwise, output "Failure".

THEOREM 3.7. *Algorithm 3.3 runs in polynomial time under ERH, and outputs "Failure" if and only if f is c-super square balanced.*

Proof. Consider first the running time under ERH. Step 0 can be done in polynomial time by Theorem 3.1. For each k, all the other steps can be finished in polynomial time: 1.1 by Huang (1991), 1.2 by Lemma 3.4, 1.3 by Lemma 3.6, 1.4 by Theorem 3.1 and Lemma 3.4, 2.1 by Huang (1991) and Lemma 3.6, 2.2 by Theorem 3.3, and 2.3 by Lemma 3.4. These steps are repeated for polynomially many values of k. Therefore Algorithm 3.3 runs in polynomial time under ERH.

Now suppose that Algorithm 3.3 outputs "Failure". We prove that f is c-super square balanced, i.e., the three conditions **(i), (ii)** and **(iii)** at the end of Section 2.4 are satisfied. Note that F_k is the set of the roots of $f^{(k)}$. The algorithm does not halt at Step 0 means that F_1 is square balanced. For each $k > 1$, Steps 1.1, 1.2, 1.3 make sure that $f^{(k)}$ has no repeated roots and its roots are different from those of f. So F_k has cardinality n and is

disjoint from F_1 for all $k > 1$. To pass Step 1.4, F_k has to be square balanced. Hence the condition **(i)** is satisfied. Step 2.1 makes sure that that $f^{(k)}$ and $f^{(\ell)}$ have no common roots, i.e., F_k is disjoint from F_ℓ, hence **(ii)** holds. Note that both A^k and A^ℓ generate the ring $\mathcal{R}$ over $\mathbb{F}_p$, as their minimal polynomials $f^{(k)}$ and $f^{(\ell)}$ over $\mathbb{F}_p$ have degree n. Hence $A^k \mapsto A^\ell$ induces an automorphism of $\mathcal{R}$. In the following, we prove that if Step 2.3 does not find a proper factor of f then **(iii)** must be satisfied.

At the start of Step 2.2, g is squarefree and completely splitting over $\mathbb{F}_p$. $F_k \cup F_\ell$ is the set of roots of g. Also, F_k and F_ℓ are both square balanced. We need to determine when the u and v computed in Step 2.2 are both trivial factors. Order the roots of $g = f^{(k)} \cdot f^{(\ell)}$ as $\eta_1, \eta_2, \ldots, \eta_{2n}$ where $\eta_i = \xi_i^\ell$ for $1 \le i \le n$ and $\eta_i = \xi_{i-n}^k$ for $n < i \le 2n$. For $1 \le i \le 2n$, let

$$t_i = \#\left\{1 \le j \le 2n : j \ne i, \sigma\big((\eta_i - \eta_j)^2\big) = \eta_i - \eta_j\right\}.$$

For $1 \le i \le n$, define

$$\begin{aligned} u_i &= \#\left\{1 \le j \le n : \sigma\big((\xi_i^\ell - \xi_j^k)^2\big) = \xi_i^\ell - \xi_j^k\right\}, \\ v_i &= \#\left\{1 \le j \le n : \sigma\big((\xi_i^k - \xi_j^\ell)^2\big) = \xi_i^k - \xi_j^\ell\right\}. \end{aligned}$$

Then, for $1 \le i \le n$,

$$\begin{aligned} t_i &= \#\left\{1 \le j \le n : j \ne i, \sigma\big((\xi_i^\ell - \xi_j^\ell)^2\big) = \xi_i^\ell - \xi_j^\ell\right\} \\ &\quad + \#\left\{1 \le j \le n : \sigma\big((\xi_i^\ell - \xi_j^k)^2\big) = \xi_i^\ell - \xi_j^k\right\} \\ &= \frac{n-1}{2} + u_i, \end{aligned}$$

and

$$\begin{aligned} t_{i+n} &= \#\left\{1 \le j \le n : \sigma\big((\xi_i^k - \xi_j^\ell)^2\big) = \xi_i^k - \xi_j^\ell\right\} \\ &\quad + \#\left\{1 \le j \le n : j \ne i, \sigma\big((\xi_i^k - \xi_j^k)^2\big) = \xi_i^k - \xi_j^k\right\} \\ &= v_i + \frac{n-1}{2}, \end{aligned}$$

since F_k and F_ℓ are square balanced.

Suppose that $u_1, \ldots, u_n$ are not equal. Then $t_1, \ldots, t_n$ are not equal. By Theorem 3.3, the roots $\eta_1 = \xi_1^\ell, \ldots, \eta_n = \xi_n^\ell$ of g, and of $f^{(\ell)}$, are separated. That is, there is a factor h in the list from Step 2.2 such that $\gcd(h, f^{(\ell)})$ is a proper factor of $f^{(\ell)}$. Similarly, if $v_1, \ldots, v_n$ are not equal then Step 2.3 will find a proper factor of $f^{(k)}$ for some h in the list. Therefore if no proper factor of $f^{(\ell)}$ or $f^{(k)}$ is found at Step 2.3 then $u_1 = \cdots = u_n$ and $v_1 = \cdots = v_n$, that is, F_k and F_ℓ are mutually square balanced. Hence **(iii)** holds.

Conversely, if **(i)**, **(ii)** and **(iii)** are satisfied, then one can see from the above proof that Algorithm 3.3 will not be able to find an proper factor of f, thus output "Failure". □

If we take $c = 6$, then Theorem 3.7 gives the result in the introduction for $p \equiv 3 \bmod 4$. The theorem suggests an interesting number theory problem on the existence of c-super square balanced subsets in $\mathbb{F}_p$. If c is large enough, then it is very likely that there are no such subsets in $\mathbb{F}_p$. We believe that $c = 6$ suffices.

Conjecture. *For any prime p and any integer $n > 1$, there are no super square balanced subsets in $\mathbb{F}_p$ of cardinality n.*

A confirmation to the conjecture implies that, under ERH, polynomials over finite fields can be factored deterministically in polynomial time.

Finally, we remark that it is possible to apply Algorithm 3.2 to extensions of $\mathcal{R} = \mathbb{F}_p[x]/(f)$ and obtain an interesting connection of the problem of factoring polynomials to a combinatorial structure called Hadamard designs. It may be possible that this approach will render the problem to combinatorial attacks. The details will be given elsewhere.

Acknowledgement. The author would like to thank Professor Joachim von zur Gathen for his encouragement and insightful discussions. Thanks also go to Professors Andrew Granville and Carl Pomerance for helpful suggestions and comments on the paper, particularly, Professor Pomerance suggested the use of super square balanced sets. Finally, the detailed comments and questions from an anonymous referee greatly improved the presentation of the paper.

References

Adleman, L., Manders, K., Miller, G. (1977). On taking roots in finite fields. In *Proc. 18th IEEE Symp. Foundations of Computer Science* (Providence, R.I., 1977) :175–178.

Bach, E. (1997). Comments on search procedures for primitive roots. *Math. Comp.*, **66**:1719–1727.

Bach, E., von zur Gathen, J., Lenstra, H. (1995). Deterministic factorization of polynomials over special finite fields. Preprint.

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

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

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

Evdokimov, S. A. (1989). Factorization of a solvable polynomial over finite fields and the generalized Riemann hypothesis. *Zapiski Nauchnyck Seminarov LOMI*, **176**:104–117.

Evdokimov, S. A. (1994). Factorization of polynomials over finite fields in subexponential time under GRH. In *Proceedings of the 1994 Algorithmic Number Theory Symposium*(Adleman, L. M., Huang, M.-D., editors), Ithaca, New York. Lecture Notes in Computer Science, Berlin, Springer-Verlag, 1994 :209–219.

von zur Gathen, J. (1987). Factoring polynomials and primitive elements for special primes. *Theoret. Computer Science*, **52**:77–89.

von zur Gathen, J., Shoup, V. (1992). Computing Frobenius maps and factoring polynomials. *Comput. Complexity*, **2**:187–224.

Huang, M. A. (1991). Generalized Riemann hypothesis and factoring polynomials over finite fields. *J. Algorithms*, **12**:464–481.

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

Mignotte, M., Schnorr, C.-P. (1988). Calcul déterministe des racines d'un polynôme dans un corps fini. *Comptes Rendus Académie des Sciences (Paris)*, **306**:467–472.

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

Pohlig, S. C., Hellman, M. E. (1978). An improved algorithm for computing logarithms over $GF(p)$ and its cryptographic significance. *IEEE-IT*, **IT-24**:106–110.

Rónyai, L. (1988). Factoring polynomials over finite fields. *J. Algorithms*, **9**:391–400.

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

Rónyai, L. (1992). Galois groups and factoring over finite fields. *SIAM J. Discrete Math.*, **5**:345–365.

Wan, D. (1996). Notes on factoring polynomials and zeta functions over finite fields. Preprint.

Wang, Y. (1959). On the least primitive root of a prime. *Acta. Math. Sinica*, **9**:432–441 (Chinese). *Scientia Sinica*, **10** (1961) :1–14 (English translation).

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