# Algebraic Problems Equivalent to Beating Exponent 3/2 for Polynomial Factorization over Finite Fields *

**Zeyu Guo[1], Anand Kumar Narayanan[2], and Chris Umans[3]**

1 **Computing and Mathematical Sciences, California Institute of Technology**
zguo@caltech.edu
2 **Computing and Mathematical Sciences, California Institute of Technology**
anandkn@caltech.edu
3 **Computing and Mathematical Sciences, California Institute of Technology**
umans@caltech.edu

## Abstract

The fastest known algorithm for factoring univariate polynomials over finite fields is the Kedlaya-Umans [13] (fast modular composition) implementation of the Kaltofen-Shoup algorithm [12, § 2]. It is randomized and takes $\widetilde{O}(n^{3/2}\log q + n\log^2 q)$ time to factor polynomials of degree $n$ over the finite field $\mathbb{F}_q$ with $q$ elements. A significant open problem is if the 3/2 exponent can be improved. We study a collection of algebraic problems and establish a web of reductions between them. A consequence is that an algorithm for any one of these problems with exponent better than 3/2 would yield an algorithm for polynomial factorization with exponent better than 3/2.

**1998 ACM Subject Classification** F.2.1 Computations in Finite Fields

**Keywords and phrases** Algorithms, Complexity, Finite Fields, Polynomials, Factorization.

**Digital Object Identifier** 10.4230/LIPIcs.MFCS.2016.47

## 1 Introduction

A recent trend in discrete algorithms has been to establish very efficient reductions between problems with polynomial time algorithms, with the intention of identifying barriers (conceptual or concrete) to improving the polynomial running time of the best known algorithms. A standard example is the problem 3-SUM, which seems to require essentially quadratic time, and which has been reduced to many other problems. More recently, the study of "fine-grained" complexity has broadened, with several connections established between central problems in discrete algorithms, and new conjectures beyond the 3-SUM conjecture entering the picture (see, e.g. [1, 2, 3, 17, 18, 22, 23, 24]).

In this paper we focus on a "barrier" in *algebraic* algorithms, that of improving the exponent 3/2 for univariate polynomial factorization and several other problems. Generally, algebraic problems have two relevant "size" parameters – $n$, and the field size $q$. It is typical for the dependence on $q$ to be polylogarithmic (it is for all of the problems we consider), and so we focus on the exponent on $n$ in this work. We find that exponent 3/2 seems to be a barrier for a number of problems. This points to a need to move beyond the so-called "baby steps giant steps" methodology which tends to give rise to the exponent 3/2 behavior.

* The authors were supported by NSF grant CCF 1423544 and a Simons Foundation Investigator grant.


The reductions in this paper can be seen as giving evidence that improving the 3/2 exponent may not be possible for these problems, but we believe that it "merely" gives evidence that this improvement requires a conceptual breakthrough (along the lines of going beyond the baby-steps giant-steps approach). Using the connections established in this paper, such a breakthrough for any one of the problems considered here would improve the exponent for all of them. In the discussion below, we use $\widetilde{O}$ to suppress $n^{o(1)}$ terms and $\log^{o(1)} q$ terms, in order to highlight the exponent on $n$ that is our main object of study. We also use the phrase "nearly linear time reduction" to mean a reduction that runs in time $\widetilde{O}(n \log q)$, and the phrase "3/2 exponent reducible" to mean the weaker connection that shows that beating exponent 3/2 for one problem implies beating exponent 3/2 for the other.

## 1.1 Algebraic problems with 3/2 exponent algorithms

We investigate the complexity of factoring a univariate polynomial over a finite field into its irreducible factors. The problem formally stated is,

- Factor: *Given a monic square free $f(x) \in \mathbb{F}_q[x]$ of degree $n$, write $f(x)$ as a product of its monic irreducible factors.*

The square free assumption is without loss of generality [14, 25]. Factor can be solved in randomized polynomial time [4] and there is an extensive line of research [5, 12, 21] leading to a randomized algorithm [13] with exponent 3/2. Surprisingly, even determining the *degree* of a single irreducible factor rapidly would be sufficient to improve the exponent of this algorithm. We formulate this problem as

- Factor Degree: *Given a monic square free $f(x) \in \mathbb{F}_q[x]$, find the degree of an irreducible factor of $f(x)$.*

and prove in § 2 that Factor is 3/2-exponent reducible to Factor Degree. That is, an algorithm for Factor Degree with exponent less than 3/2 yields one for Factor. Observe that Factor Degree merely seeks one, not necessarily all, irreducible factor degrees. We next investigate two linear algebraic problems, both we will demonstrate to be nearly linear time reducible to Factor.

- Frobenius Min-Poly: *Given a monic square free $f(x) \in \mathbb{F}_q[x]$, compute the minimal polynomial of the Frobenius endomorphism on $\mathbb{F}_q[x]/(f(x))$ which takes $a(x) \mod f(x)$ to $a(x)^q \mod f(x)$.*
- Carlitz Char-Poly: *Given a monic square free $f(x) \in \mathbb{F}_q[x]$, compute the characteristic polynomial of the Carlitz endomorphism on $\mathbb{F}_q[x]/(f(x))$ which takes $a(x) \mod f(x)$ to $xa(x) + a(x)^q \mod f(x)$.*

In § 4, we prove that Factor Degree is nearly linear time reducible to Carlitz Char-Poly, under certain restrictions on the characteristic of $\mathbb{F}_q$. These restrictions were removed in [15] by passing from Carlitz to Drinfeld modules. In § 3, through a novel recursive argument, we prove that Factor is 3/2-exponent reducible to Frobenius Min-Poly.

Frobenius Min-Poly was known [12, 11] to be nearly linear time reducible to

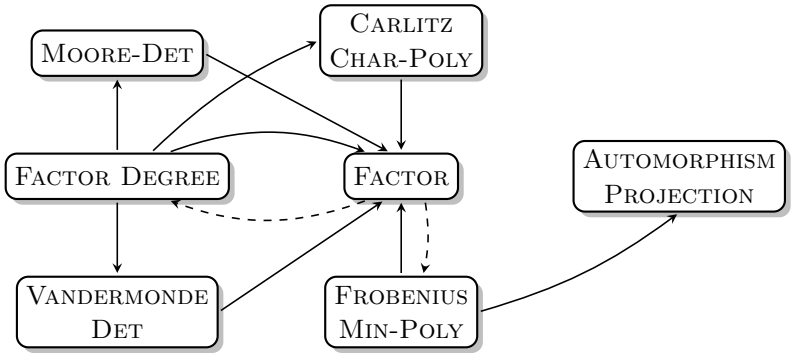
- Automorphism Projection: *Given a monic square free $f(x) \in \mathbb{F}_q[x]$, $\alpha \in \mathbb{F}_q[x]/(f(x))$ and an $\mathbb{F}_q$-linear map $u : \mathbb{F}_q[x]/(f(x)) \longrightarrow \mathbb{F}_q$, compute $u(\alpha^{q^i}), \forall i \in \{1, 2, \ldots, \deg(f)\}$.*

Thus, as a consequence of the reduction in § 3, we conclude that Factor is 3/2-exponent reducible to Automorphism Projection. This should be contrasted with the connection established in [12, 11]. They show that Factor is nearly linear time reducible to Automorphism Projection *assuming an $\mathbb{F}_q$-linear straight line program algorithm* for Automorphism Projection. This assumption allows them to use the "transpose" problem Automorphism Evaluation. Our reduction to Automorphism Projection is novel, direct, and holds without any assumptions.

The final two problems pertain to zero testing Moore and Vandermonde determinants.

- MOORE-DET: *Given a monic square free $f(x) \in \mathbb{F}_q[x]$ and a positive integer $m$, decide if the determinant of the $m$ by $m$ square matrix with entries $m_{ij} := x^{jq^i} \mod f(x)$ is zero.*
- VANDERMONDE-DET: *Given a monic square free $f(x) \in \mathbb{F}_q[x]$ and a positive integer $b \leq \sqrt{\deg(f)}$, decide if the determinant of the Vandermonde matrix with first row $(x^{q^i} \mod f(x), i = 0, 1, 2, \ldots, b-1, b, 2b, 3b, \ldots, (b-1)b, b^2)$ is zero.*

In § 5, we prove that FACTOR DEGREE is nearly linear time reducible to each of these problems and that each of these problems is nearly linear time reducible to FACTOR. In summary, we have the following diagram where solid lines denote nearly linear time reductions and dotted lines denote 3/2-exponent reductions.




An interesting open question is if the dotted lines can be made solid. Except for AUTOMORPHISM PROJECTION, every listed problem has a known randomized algorithm with exponent 3/2. If the matrix multiplication exponent is 2, then a randomized algorithm for AUTOMORPHISM PROJECTION with exponent 3/2 is known. Another open problem is if this dependence on the matrix multiplication exponent can be removed – perhaps by reducing AUTOMORPHISM PROJECTION to one of the other problems in the figure. Regardless, an algorithm for any of the problems in the figure with exponent less that 3/2 would yield an algorithm with exponent 3/2 for FACTOR, and this is one of the main points of this paper.

## 2 Factorization and Finding a Factor Degree

Clearly, if one can solve the problem FACTOR in time $T(n, q)$ then one can solve the problem FACTOR DEGREE in time $T(n, q)$. In this section we show a reduction in the reverse direction, which leads to the surprising conclusion that one *only* needs to compute the *degree* of a single irreducible factor of the polynomial $f(x)$ with exponent better than 3/2 to be able to factor $f(x)$ completely with exponent better than 3/2.

▶ **Theorem 1.** *If there is an algorithm that solves* FACTOR DEGREE *in the time $T(n, q)$ where $T(n, q) = \Omega(n \log^2 q)$* [1]*, then there is an algorithm that solves* FACTOR *in time $\widetilde{O}(n \cdot T(n, q)^{1/3} \log^{4/3} q)$.*

Observe that when FACTOR DEGREE has an exponent 3/2 algorithm (as it does), this reduction recovers a 3/2 exponent algorithm for FACTOR. A sub-3/2 exponent algorithm for FACTOR DEGREE implies a sub-3/2 exponent algorithm for FACTOR, with a nearly-linear time algorithm yielding exponent 4/3 for FACTOR.

[1] The assumption $T(n, q) = \Omega(n \log^2 q)$ is without loss of generality. For otherwise we slow down an algorithm with runtime $T(n, q)$ until it is $\Omega(n \log^2 q)$.

**Proof.** We are given a monic, square-free polynomial $f(x) \in \mathbb{F}_q[x]$ of degree $n$. Let $g(x)$ be the product of irreducible factors of $f(x)$ with degrees at most $t$ (for a parameter $t$ to be chosen later). If $s(x)$ is defined as $s(x) = \prod_{i=1}^{t}(x^{q^i} - x)^{a_i}$, for some positive integers $a_1, a_2, \ldots, a_t$, then we have that $g(x) = \gcd(s(x), f(x))$. Using fast modular composition [13] and the method of Kaltofen-Shoup [12], we can compute $s(x) \bmod f(x)$ in time $\widetilde{O}(n\sqrt{t}\log^2 q)$ time. We then proceed to factor $g(x)$ completely, using the Kedlaya-Umans implementation of the Kaltofen-Shoup algorithm. The bottleneck in this algorithm is computing the splitting polynomials, which are all polynomials of the form of $s(x)$, with $i$ ranging from 1 up to $t' \leq t$. This portion of the algorithm runs in time $\widetilde{O}(n\sqrt{t}\log^2 q)$ and factors $g(x)$ completely.

Now we invoke the algorithm to solve FACTOR DEGREE, on input $f(x)/g(x)$. Upon finding the degree $d$ of an irreducible factor, we compute $\gcd(x^{q^d} - x \bmod f(x), f(x))$ to split off the factors with that degree. We then repeat. The number of repetitions is bounded by $n/t$, since each irreducible factor of $f(x)/g(x)$ has degree at least $t$. Each repetition takes time $T(n,q) + \widetilde{O}(n\log^2 q)$. Thus this portion of the algorithm runs in time $\widetilde{O}(n/t \cdot (T(n,q) + n\log^2 q)) = \widetilde{O}(n/t \cdot T(n,q))$. Finally we factor completely using equal-degree factorization which takes $\widetilde{O}(n\log^2 q)$ time. Optimizing, we set $t = (T(n,q)/\log^2 q)^{2/3}$, and the overall running time becomes $\widetilde{O}(n \cdot T(n,q)^{1/3}\log^{4/3} q)$ for each of the two stages, and hence in total as well. ◀

## 3 Factoring and Minimal Polynomial of Frobenius

For a monic square free $f(x)$, let $g(\lambda) \in \mathbb{F}_q[\lambda]$ denote the minimal polynomial of the $q^{th}$ power Frobenius endomorphism $\sigma : \mathbb{F}_q[x]/(f(x)) \to \mathbb{F}_q[x]/(f(x))$. That is, $g(\lambda)$ is the unique nonzero monic polynomial of least degree such that the endomorphism $g(\sigma)$ on $\mathbb{F}_q[x]/(f(x))$ is zero. The problem FROBENIUS MIN-POLY is to determine $g(\lambda)$ given $f(x)$. Since $g(\lambda)$ is the least common multiple of $\lambda^d - 1$ as $d$ runs through the degrees of the irreducible factors of $f(x)$, FROBENIUS MIN-POLY is nearly linear time reducible to FACTOR. In this section, we conversely prove that FACTOR is 3/2-exponent reducible to FROBENIUS MIN-POLY.

Let `FrobMinPoly` be an oracle that solves FROBENIUS MIN-POLY. We present an algorithm `Factor` that invokes `FrobMinPoly` and solves FACTOR. For $k \in \mathbb{N}^+$, denote by $\Phi_k$ the $k$th cyclotomic polynomial over $\mathbb{F}_q$. Write $\phi(\cdot)$ for the Euler totient function.

**Algorithm 1** `Factor`$(f(x))$

**Input:** Monic square free polynomial $f(x) \in \mathbb{F}_q[x]$ of degree $n$.
**Output:** Monic irreducible factors of $f(x)$.
**Oracle:** `FrobMinPoly`

1: Using [13], output and remove all monic irreducible factors of $f(x)$ of degree at most $n^{2/3}$. If at most one irreducible factor of degree greater than $n^{2/3}$ remains, output and exit.
2: $g(\lambda) \leftarrow$ `FrobMinPoly`$(f(x))$.
3: Perform square free factorization on $g(\lambda)$, and then run `Factor` recursively on the outputs to obtain the list of monic irreducible factors $g_1(\lambda), \ldots, g_m(\lambda)$ of $g(\lambda)$.
4: Run `FindT`$(g_1(\lambda), \ldots, g_m(\lambda))$ computing the set $T := \{k : p \nmid k \text{ and } \Phi_k(\lambda) | g(\lambda)\}$ as well as $m_k$, the multiplicity of $\Phi_k(\lambda)$ in $g(\lambda)$, for each $k \in T$.
5: Compute $S := \{kp^e : k \in T, 0 \leq e \leq \log_p m_k\}$.
6: **for each** $s \in S$ greater than $n^{2/3}$, set $f_s(x) \leftarrow \gcd(f(x), x^{q^s} - x \bmod f(x))$, $f(x) \leftarrow f(x)/f_s(x)$ and perform equal-degree factorization on $f_s(x)$.

The algorithm begins by extracting all monic irreducible factors of degree at most $n^{2/3}$. After Line 1, $f(x)$ only has large (at least $n^{2/3}$) degree factors. Suppose $d_1, d_2, \ldots, d_m$ are the degrees of the (remaining) monic irreducible factors of $f(x)$. Then the minimal polynomial $g(\lambda) \in \mathbb{F}_q[\lambda]$ of the Frobenius acting on $\mathbb{F}_q[x]/(f(x))$ is

$$g(\lambda) = \operatorname{lcm}\left(\lambda^{d_1} - 1, \ldots, \lambda^{d_m} - 1\right).$$

In particular, the cyclotomic polynomials $\Phi_{d_1}(\lambda), \ldots, \Phi_{d_m}(\lambda)$ divide $g(\lambda)$ and the factorization of $g(\lambda)$ contains information about $d_1, d_2, \ldots, d_m$. We devise a novel procedure to infer $d_1, d_2, \ldots, d_m$ efficiently.

On Line 2, $g(\lambda)$ is computed by invoking `FrobMinPoly`. To infer $d_1, d_2, \ldots, d_m$, we seek the factorization of $g(\lambda)$. To this end, a key idea is to factor $g(\lambda)$ recursively on Line 3 and obtain a list $g_1(\lambda), g_2(\lambda), \ldots, g_m(\lambda)$ of its monic irreducible factors. Since $f(x)$ is not irreducible at this point, $g(\lambda)$ has degree strictly less than $f(x)$ and the algorithm runs to completion.

Then we use a procedure `FindT`$(g_1(\lambda), \ldots, g_m(\lambda))$ to compute the set $T$ and integers $m_k$ as defined on Line 4. This step is the most technical part of the algorithm, and we defer its description and analysis to the next subsection, where we prove the following theorem:

▶ **Theorem 2.** `FindT`$(g_1(\lambda), \ldots, g_m(\lambda))$ *can be implemented to run in* $\widetilde{O}(n \log q)$ *time.*

Once $T$ is known, to compute $S$ on Line 5 is straightforward. The following lemma shows that $S$ indeed contains $d_1, d_2, \ldots, d_m$.

▶ **Lemma 3.** $d_1, d_2, \ldots, d_m \in S$.

**Proof.** Consider an arbitrary $d \in \{d_1, d_2, \ldots, d_m\}$ and write it as $d = kp^e$ with $p \nmid k$. By definition $(\lambda^d - 1) | g(\lambda)$. Since $\lambda^d - 1 = (\lambda^k - 1)^{p^e}$ and $\lambda^k - 1 = \prod_{k_0 | k} \Phi_{k_0}(\lambda)$, $\Phi_k(\lambda)$ is a factor of $g(\lambda)$ with multiplicity at least $p^e$. So $k \in T$ and $m_k \geq p^e$, implying $d = kp^e \in S$. ◀

To conclude, by Line 6, all the irreducible factors of $f(x)$ are indeed output.

▶ **Theorem 4.** *Suppose the oracle* `FrobMinPoly` *runs in time* $T(n, q)$ *which is monotone in* $n$ *and* $q$. *Then* `Factor` *factors a degree-*$n$ *polynomial in* $\widetilde{O}(T(n, q) + n^{4/3} \log^2 q)$ *time.*

**Proof.** We first analyze the running time of each step except the recursive call. Line 1 can be implemented in $\widetilde{O}(n^{4/3} \log^2 q)$ time using the baby-step-giant-step strategy [12, 13]. The oracle `FrobMinPoly` on Line 2 runs in time $T(n, q)$. The set $T$ on Line 4 could be found in time $\widetilde{O}(n \log q)$ by Theorem 2. Since $\prod_{k \in T} \Phi_k(\lambda)^{m_k}$ divides $g(\lambda)$, we have $\sum_{k \in T} m_k \phi(k) \leq \deg(g(\lambda)) \leq n$. Hence $|T| \leq n$ and $m_k \leq n$ for all $k \in T$, implying $S$ on Line 5 could be computed in time $\widetilde{O}(n)$. Further,

$$\sum_{s \in S} s \leq \sum_{k \in T, 0 \leq e \leq \log_p m_k} kp^e \leq \log n \sum_{k \in T} k m_k \leq O(\log \log n) \cdot \log n \sum_{k \in T} m_k \phi(k) = \widetilde{O}(n)$$

where we use $k/\phi(k) = O(\log \log k)$ [19] and $\sum_{k \in T} m_k \phi(k) \leq n$. Hence the number of $s \in S$ greater than $n^{2/3}$ is at most $(\sum_{s \in S} s)/n^{2/3} = \widetilde{O}(n^{1/3})$. For each $s \in S$, Computing $f_s(x)$ takes $\widetilde{O}(n \log^2 q)$ time for each $s \in S$ [13] and hence $\widetilde{O}(n^{4/3} \log^2 q)$ time in total. Equal degree factorization on Line 6 takes $\widetilde{O}(n \log^2 q)$ time in total.

Let $d_{\max}(f(x))$ denote the maximal degree of the irreducible factors of $f(x)$. We claim that $d_{\max}(f(x))$ shrinks by at least a factor of two every two recursive calls. It implies that the recursive tree has depth no more than $O(\log n)$, so the total running time is bounded by $O(\log n) \cdot (T(n, q) + \widetilde{O}(n^{4/3} \log^2 q)) = \widetilde{O}(T(n, q) + n^{4/3} \log^2 q)$, as desired.

Consider an irreducible factor $g_0(\lambda)$ of $g(\lambda)$. We know $g_0(\lambda)$ divides $\lambda^k - 1 = \prod_{k_0|k} \Phi_{k_0}(\lambda)$ for a positive integer $k$ corresponding to some degree $k$ irreducible factor $f_0(x)$ of $f(x)$. If $g_0(\lambda)$ divides $\Phi_{k_0}(\lambda)$ for some proper divisor $k_0$ of $k$, we have $\deg(g_0(\lambda)) \leq \phi(k_0) \leq k_0 \leq k/2$. Likewise, if $g_0(\lambda)$ is a proper irreducible factor of $\Phi_k(\lambda)$, we have $\deg(g_0(\lambda)) \leq \phi(k)/2 \leq k/2$ as well. So assume $g_0(\lambda) = \Phi_k(\lambda)$. Suppose $k = \prod_\ell \ell^{e_\ell}$, $\ell$ running over prime divisors of $k$. Then $\phi(k) = \prod_\ell (\ell - 1)\ell^{e_\ell - 1}$. If $k$ is even, we have $e_2 \geq 1$ implying $\deg(g_0(\lambda)) = \phi(k) \leq k/2$ (since for $\ell = 2$, $(\ell - 1)\ell^{e_\ell - 1} = \ell^{e_\ell}/2$). If $k$ is odd, $\deg(g_0(\lambda)) = \phi(k) = \prod_\ell (\ell - 1)\ell^{e_\ell - 1}$ is even. The argument above applied to $g_0(\lambda)$ and $g(\lambda)$ in place of $f_0(x)$ and $f(x)$ shows that the degree shrinks by at least a factor of two in the next recursive call. The claim follows. ◀

▶ Remark. One may easily check that the same algorithm and analysis also work if the polynomial $g(\lambda)$ computed by the oracle is the *characteristic polynomial* of the Frobenius endomorphism instead of the minimal polynomial. The only difference is that $g(\lambda)$ is the product of $\lambda^{d_1} - 1, \ldots, \lambda^{d_m} - 1$ rather than their lcm.

## 3.1 Computing the Set $T$

We next devise a nearly linear time procedure to implement `FindT`. It relies on solutions to the following two problems: (1) finding all irreducible factors of $\Phi_k(\lambda)$ over $\mathbb{F}_q$ from a single irreducible factor $g_0(\lambda)$ and (2) finding the corresponding integer $k$. We deal with these two problems individually before describing `FindT`.

### 3.1.1 Finding the irreducible factors of $\Phi_k(\lambda)$

Let $k \in [1, n]$ be an integer coprime to $p$. Our goal is to find all the irreducible factors of $\Phi_k(\lambda)$ over $\mathbb{F}_q$ from a single irreducible factor $g_0(\lambda) | \Phi_k(\lambda)$. To achieve it, we need to know how $\Phi_k(\lambda)$ factorizes over $\mathbb{F}_q$.

#### 3.1.1.1 Factorization of $\Phi_k(\lambda)$ over $\mathbb{F}_q$.

As $k$ is coprime to $p$, there are $\phi(k)$ distinct primitive $k$th roots of unity in $\overline{\mathbb{F}}_q$ which are exactly the roots of $\Phi_k(\lambda)$. Denote this set of roots by $\mu_k$. Let $G$ be the abelian group $(\mathbb{Z}/k\mathbb{Z})^\times$ of order $\phi(k)$. For $d \in \mathbb{Z}$, we write $\bar{d}$ for the image of $d$ in $\mathbb{Z}/k\mathbb{Z}$. The group $G$ acts on $\mu_k$ such that $\bar{d} \in G$ sends any $\theta \in \mu_k$ to $\theta^d$. This is a regular action, meaning that for fixed $\theta \in \mu_k$, the map $\bar{d} \mapsto \theta^d$ is a bijection between $G$ and $\mu_k$. As $p$ is coprime to $k$, we have $\bar{q} \in G$. Let $G_0 = \langle \bar{q} \rangle \subseteq G$ and $s = [G : G_0]$. Restrict the $G$-action on $\mu_k$ to a $G_0$-action. Then $\mu_k$ is partitioned into $s$ distinct $G_0$-orbits represented by $\theta_1, \ldots, \theta_s \in \mu_k$. It is well-known that the factorization of $\Phi_k(\lambda)$ over $\mathbb{F}_q$ is then determined in the following way:

▶ **Lemma 5.** *Under the notations above, $\Phi_k(\lambda)$ has $s$ irreducible factors $g_1(\lambda), \ldots, g_s(\lambda)$ over $\mathbb{F}_q$ corresponding to the $G_0$-orbits $G_0\theta_1, \ldots, G_0\theta_s$ of $\mu_k$ in the sense that the set of roots of $g_i(\lambda)$ is exactly $G_0\theta_i$.*

**Proof.** Let $g(\lambda)$ be an irreducible factor of $\Phi_k(\lambda)$ over $\mathbb{F}_q$ and $\theta \in \mu_k$ be a root of $g(\lambda)$. Then $\mathbb{F}_q[\theta]$ is Galois over $\mathbb{F}_q$ with the Galois group generated by the Frobenius map $a \mapsto a^q$. So $a \in \mathbb{F}_q[\theta]$ is a root of $g(\lambda)$ if and only if $a^q$ is a root of $g(\lambda)$. Therefore $G_0\theta$ is the set of roots of $g(\lambda)$ and the lemma follows. ◀

From now on we fix a root $\theta \in \mu_k$ of the given irreducible factor $g_0$ of $\Phi_k$. For any subgroup $H \subseteq G$ containing $G_0$, the $G$-action on $\mu_k$ restricts to an $H$-action. The $H$-orbit $H\theta$ is partitioned into a disjoint union of $G_0$-orbits and hence corresponds to a subset $L$ of

irreducible factors of $\Phi_k(\lambda)$ by Lemma 5. Note that $L$ also determines $H$: $h \in G$ lies in $H$ if and only if the minimal polynomial of $h\theta$ over $\mathbb{F}_q$ is in $L$. We say $L$ is *associated with* the subgroup $H$.

We need an auxiliary procedure $\texttt{FindOrder}(\ell, L)$ to find the order of $H\bar{\ell}$ in $G/H$.

▶ **Lemma 6.** *There exists a procedure* $\texttt{FindOrder}(\ell, L)$ *that takes an integer* $\ell$ *and the set* $L$ *associated with* $H$*, and returns the following result: if* $\bar{\ell} \in G = (\mathbb{Z}/k\mathbb{Z})^\times$*, it returns the order of* $H\bar{\ell}$ *in* $G/H$*, i.e. the smallest* $e > 1$ *for which* $\bar{\ell}^e \in H$*. Otherwise it returns zero. Moreover* $\texttt{FindOrder}(\ell, L)$ *could be implemented in time* $\widetilde{O}(\phi(k) \log q)$*.*

See the full version of this paper [10] for the proof of Lemma 6 and the description of $\texttt{FindOrder}(\ell, L)$.

We use a randomized procedure $\texttt{FindCyclotomic}(g_0(\lambda), n)$ to find all irreducible factors of $\Phi_k(\lambda)$ over $\mathbb{F}_q$. Here $g_0(\lambda)$ is one irreducible factor of $\Phi_k(\lambda)$ and $n$ is the degree of the polynomial $f(x)$.[2]

**Algorithm 2** $\texttt{FindCyclotomic}(g_0(\lambda), n)$

**Input:** Irreducible factor $g_0(\lambda)$ of $\Phi_k(\lambda)$ over $\mathbb{F}_q$ and degree $n$ of $f(x)$
**Output:** The list of irreducible factors of $\Phi_k(\lambda)$ over $\mathbb{F}_q$

1: $L \leftarrow \{g_0(\lambda)\}$
2: **for** $t$ **from** 1 **to** $N = \lfloor c \log n \log\log n \rfloor$ **do** ▷ $c > 0$ *is a large enough constant*
3: Pick an integer $\ell \in [1, n]$ at random
4: $e \leftarrow \texttt{FindOrder}(\ell, L)$
5: **for each** $h(\lambda) \in L$ **do**
6: $r_0 \leftarrow \lambda \bmod h(\lambda) \in \mathbb{F}_q[\lambda]/(h(\lambda))$, $r_i \leftarrow r_{i-1}^\ell$ for $i = 1, \ldots, e-1$
7: Let $f_i(\lambda)$ be the minimal polynomial of $r_i$ over $\mathbb{F}_q$ for $i = 1, \ldots, e-1$
8: Add $f_1(\lambda), \ldots, f_{e-1}(\lambda)$ to $L$
9: **end for**
10: **end for**
11: **return** $L$

The procedure $\texttt{FindCyclotomic}(g_0(\lambda), n)$ maintains a subset $L$ of irreducible factors of $\Phi_k(\lambda)$ associated with some subgroup of $G$ containing $G_0$. Initially $L = \{g_0(\lambda)\}$, associated with $H_0 := G_0$. We claim:

▶ **Lemma 7.** *Suppose* $L$ *is associated with* $H_{i-1}$ *at the beginning* $i$*th execution of the outer loop of* $\texttt{FindCyclotomic}(g_0(\lambda), n)$*. Then at the end of the* $i$*th execution, the set* $L$ *is associated with a subgroup* $H_i \supseteq H_{i-1}$*. Moreover,* $H_i = H_{i-1}$ *if* $\bar{\ell} \notin G$ *in the* $i$*th execution of the outer loop. Otherwise* $H_i = H_{i-1}\langle\bar{\ell}\rangle$*.*

**Proof.** If $\ell \notin G$ in the $i$th execution of the outer loop, then $e$ is set to zero by Lemma 6 and the claim is trivial. So assume $\ell \in G$ and let $H = H_{i-1}\langle\bar{\ell}\rangle$. Then $e$ is the order of $H_{i-1}\bar{\ell}$ in $H/H_{i-1}$ by Lemma 6, or $[H : H_{i-1}]$. Suppose the irreducible factors in $L$ at the beginning of the $i$th execution correspond to distinct $G_0$-orbits $G_0\theta_1, \ldots, G_0\theta_m$ whose union is the $H_{i-1}$-orbit $H_{i-1}\theta$, $m = [H_{i-1} : G_0]$. The inner loop enumerates $G_0\theta_j$, and for each of them, adds the irreducible factor corresponding to $G_0\theta_j^{\ell^s}$ to $L$, $s = 1, \ldots, e-1$. Note that the union of these $G_0$-orbits $G_0\theta_j^{\ell^s} = G_0\bar{\ell}^s\theta_j = \bar{\ell}^s G_0\theta_j$ where $1 \le j \le m$, $0 \le s \le e-1$ equals

[2] The argument $n$ is only used on Line 2 and 3 to control the number of repetitions and the range of $\ell$, which is related to the error probability.

the union of $H_{i-1}$-orbits $\bar{\ell}^s H_{i-1}\theta_j$, which equals the $H$-orbit $H\theta$. And these $G_0$-orbits are all distinct since the number of them is $me = [H : G_0]$. So $L$ is associated with $H$ at the end of the $i$th execution of the outer loop. ◀

▶ **Lemma 8.** *The procedure* `FindCyclotomic`$(g_0(\lambda), n)$ *returns a set $L$ associated with $H_N \subseteq G$. And $H_N = G$ with probability $1 - \mathrm{poly}(n)$ in which case $L$ contains all irreducible factors of $\Phi_k(\lambda)$ over $\mathbb{F}_q$. Moreover* `FindCyclotomic`$(g_0(\lambda), n)$ *could be implemented in time $\widetilde{O}(\phi(k) \log q)$.*

**Proof.** We want to show $H_N = G$ with probability $1 - \mathrm{poly}(n)$. By Lemma 6 and Lemma 7, it is equivalent to showing the set of $\bar{\ell} \in G$ generates $G$. Identify $G$ with a product of at most $\log |G| \le \log n$ primary cyclic groups $C_i$ whose orders are coprime to each other. We only need to show the the set of holomorphic images of $\bar{\ell} \in G$ generates $C_i$ for each $i$ with probability $1 - \mathrm{poly}(n)$ and then apply the union bound.

So fix one such $C_i$ and let $m = |C_i|$. Then $\phi(m)$ out of the $m$ elements in $C_i$ are generators of $C_i$. Let $\alpha$ be the probability that the holomorphic image of $\bar{\ell}$ is among these $\phi(m)$ elements, where $\ell$ is randomly sampled from $[1, n]$ as on Line 3. As $m$ is a prime power, we have $\phi(m) \ge m/2$. Therefore

$$\alpha \ge \frac{\lfloor n/k \rfloor}{n} \cdot \frac{\phi(m)}{m} \cdot |G| = \Omega(\phi(k)/k) = \Omega(1/\log\log k)$$

where we use $k/\phi(k) = O(\log\log k)$ [19]. So for sufficiently large $N = \lfloor c \log n \log\log n \rfloor$, the claim holds with probability $1 - \mathrm{poly}(n)$.

Then we analyze the running time: Line 4 runs in time $\widetilde{O}(\phi(k) \log q)$ by Lemma 6. Line 7 could be implemented in time $\widetilde{O}(|G_0| \log q)$ [13, 20]. And Line 3–9 runs in time $|L| \cdot \max\{e, 1\} \cdot \widetilde{O}(|G_0| \log q)$. This is bounded by $\widetilde{O}(\phi(k) \log q)$ since $|L| = [H_{i-1} : G_0]$, $\max\{e, 1\} = [H_i : H_{i-1}]$, and $|G| = \phi(k)$. As $N = \Theta(\log n \log\log n)$, the total running time is bounded by $\widetilde{O}(\phi(k) \log q)$. ◀

### 3.1.2 Finding the integer $k$

Another problem we need to solve is finding the integer $k$ given an irreducible factor $g_0(\lambda)$ of $\Phi_k(\lambda)$ over $\mathbb{F}_q$. Using the procedure `FindCyclotomic`$(g_0(\lambda))$, we could find all the irreducible factors of $\Phi_k(\lambda)$ and hence $\Phi_k(\lambda)$ itself. The degree $d := \deg(\Phi_k(\lambda)) = \phi(k)$ is hence also known. If $|\phi^{-1}(d)|$ is small, we could find $k$ by enumerating $k_0 \in \phi^{-1}(d)$ and checking if $\Phi_{k_0}(\lambda) = \Phi_k(\lambda)$. However, Erdős [9] showed that for some constant $c > 0$, there are infinitely many integers $d$ for which $|\phi^{-1}(d)| \ge d^c$. So this approach is not affordable in general. Based on more sophisticated ideas, we show that it is possible to find $k$ efficiently:

▶ **Lemma 9.** *There exists a procedure* `Findk`$(d, g_0(\lambda))$ *that takes an integer $d > 0$ dividing $\phi(k)$ and an irreducible factor $g_0(\lambda)$ of $\Phi_k(\lambda)$, and returns a positive integer $k_0 | k$ in time $\widetilde{O}(\phi(k) \log q)$. Moreover $k_0 = k$ if $d = \phi(k)$.*

See the full version of this paper [10] for the proof of Lemma 9 and the description of `Findk`$(d, g_0(\lambda))$.

### 3.1.3 Finding the set $T$

Now we are ready to describe the procedure `FindT`$(g_1(\lambda), \ldots, g_m(\lambda))$:

**Algorithm 3** `FindT`$(g_1(\lambda), \ldots, g_m(\lambda))$

**Input:** The irreducible factors $g_1(\lambda), \ldots, g_m(\lambda)$ of $g(\lambda)$ over $\mathbb{F}_q$
**Output:** The set $T$ and multiplicities $m_k$ for each $k \in T$

1: $L_0 \leftarrow \{g_1(\lambda), \ldots, g_m(\lambda)\}$ as a multi-set and $T \leftarrow \emptyset$
2: **repeat**
3: Pick an arbitrary element $g_0(\lambda) \in L_0$
4: $L \leftarrow$ `FindCyclotomic`$(g_0(\lambda))$
5: $h(\lambda) \leftarrow \prod_{f_i(\lambda) \in L} f_i(\lambda)$, $d \leftarrow \deg(h(\lambda))$
6: $k_0 \leftarrow$ `Findk`$(d, g_0)$
7: **if** $h(\lambda) | \lambda^{k_0} - 1$ **then**
8: **if** $k_0 \notin T$ then $m_{k_0} \leftarrow 0$
9: $T \leftarrow T \cup \{k_0\}$, $m_{k_0} \leftarrow m_{k_0} + 1$
10: $L_0 \leftarrow L_0 - L$
11: **end if**
12: **until** $L_0 = \emptyset$
13: **return** $T$

▶ **Theorem 10** (Theorem 2 restated). `FindT`$(g_1(\lambda), \ldots, g_m(\lambda))$ *computes the set* $T$ *and multiplicities* $m_k$ *as defined in Algorithm 1, Line* 4. *Moreover it halts in time* $\widetilde{O}(n \log q)$ *with probability* $1 - 1/\mathrm{poly}(n)$.

**Proof.** The algorithm picks $g_0(\lambda)$ from $L_0$, calls `FindCyclotomic` to find a list $L \subseteq L_0$ that almost surely contains all the irreducible factors of $\Phi_k(\lambda)$, and remove these factors from $L_0$. It repeats these steps until $L_0$ is empty. Each time it also determines the integer $k$ using `Findk`, adds it to $T$ and updates $m_k$.

Note that with small probability, the list $L$ returned by `FindCyclotomic` may not contain all the irreducible factors, in which case it is associated with a proper subgroup $H_N \subseteq G$ (c.f. Lemma 8). In any case we have $\deg(h(\lambda)) | \phi(k)$ and therefore by Lemma 9, the integer $k_0$ returned by `Findk` divides $k$. We verify that $k = k_0$ on Line 7: $h(\lambda) | (\lambda^{k_0} - 1)$ if and only if $k | k_0$ if and only if $k = k_0$ since we know $k_0 | k$. And if we find $k \neq k_0$ we do nothing in that round. The correctness of the algorithm is then straightforward.

For the running time, note that each round runs in time $\widetilde{O}(\phi(k) \log q)$ by Lemma 8 and Lemma 9, and then factors of total degree $\phi(k)$ are removed from $L_0$ with probability $1 - 1/\mathrm{poly}(n)$. So with probability $1 - 1/\mathrm{poly}(n)$, the total running time is bounded by $\sum_{i=1}^{m} \widetilde{O}(\deg(g_i(\lambda)) \log q) = \widetilde{O}(n \log q)$. ◀

## 4 Polynomial Factorization Using Carlitz Modules

We next establish connections between polynomial factorization and the Carlitz action. We prove two nearly linear reductions, namely Factor Degree to Carlitz Char-Poly and Carlitz Char-Poly to Factor. The former reduction requires that the characteristic $p$ of $\mathbb{F}_q$ is larger than the number of irreducible factors.

### 4.1 Carlitz Modules

Let $A$ be an $\mathbb{F}_q[x]$-algebra. For $f(x) \in \mathbb{F}_q[x]$ and $\alpha \in A$, $f(x)\alpha$ is understood to be the result of the $\mathbb{F}_q[x]$ action of $f(x)$ on $\alpha$ in $A$. Let $\sigma : A \longrightarrow A$ and $\tau : A \longrightarrow A$ denote the $q^{th}$ power Frobenius endomorphism and the multiplication by $x$ endomorphism respectively. That is,

$\forall \alpha \in A$, $\sigma(\alpha) = \alpha^q$ and $\tau(\alpha) = x\alpha$. In [6, 7], Carlitz endowed a new $\mathbb{F}_q[x]$-module structure on $A$ by defining $m(x) = \sum_i m_i x^i \in \mathbb{F}_q[x]$ to act on $\alpha \in A$ as

$$\rho_m(\alpha) := (m(\sigma + \tau))(\alpha) = \left( \sum_i m_i (\sigma + \tau)^i \right)(\alpha).$$

In particular, $\forall \alpha \in A$, $\rho_x(\alpha) = \alpha^q + x\alpha$ and $\forall u \in \mathbb{F}_q$, $\rho_u(\alpha) = u\alpha$. Let $\rho(A)$ denote the $\mathbb{F}_q[x]$-module structure thus endowed to $A$ by the Carlitz action. To factor a monic square free polynomial $f(x)$, we will concern ourselves with $\rho(\mathbb{F}_q[x]/(f(x)))$. Let $\chi_f(x) \in \mathbb{F}_q[x]$ denote the characteristic polynomial of the $\mathbb{F}_q$ linear transformation on $\mathbb{F}_q[x]/(f(x))$ that takes $\alpha \in \mathbb{F}_q[x]/(f(x))$ to $\rho_x(\alpha)$. Hence Carlitz Char-Poly may be restated as

▶ **Problem 11.** Given a monic square free $f(x) \in \mathbb{F}_q[x]$, compute $\chi_f(x)$.

By Lemma 12, knowledge of factorization of $f(x)$ immediately yields $\chi_f(x)$ in $\widetilde{O}(n \log q)$ time. Thus Problem Carlitz Char-Poly is linear time reducible to Factor. We next reduce Factor Degree to Carlitz Char-Poly.

## 4.2 Factor Degree Estimation using Carlitz Modules

▶ **Lemma 12.** *Let $f(x) = \prod_i f_i(x)$ be a factorization of a monic square free $f(x) \in \mathbb{F}_q[x]$ into monic irreducible polynomials . Then $\rho(\mathbb{F}_q[x]/(f(x))) \cong \bigoplus_i \mathbb{F}_q[x]/(f_i(x) - 1)$. In particular, $\chi_f(x) = \prod_i (f_i(x) - 1)$.*

**Proof.** By the Chinese remainder theorem, $\mathbb{F}_q[x]/(f(x)) \cong \prod_i \mathbb{F}_q[x]/(f_i(x))$

$$\Rightarrow \rho(\mathbb{F}_q[x]/(f(x))) \cong \rho\left( \prod_i \mathbb{F}_q[x]/(f_i(x)) \right) \cong \bigoplus_i \rho\left(\mathbb{F}_q[x]/(f_i(x))\right). \tag{4.1}$$

The final congruence holds since for every direct product $C \cong A \times B$ of $\mathbb{F}_q[x]$-algebras, we have the corresponding direct sum $\rho(C) \cong \rho(A) \oplus \rho(B)$ of $\mathbb{F}_q[x]$-modules [8]. For a monic irreducible $g(x)$ ([8]),

$$\rho(\mathbb{F}_q[x]/(g(x))) \cong \mathbb{F}_q[x]/(g(x) - 1). \tag{4.2}$$

Equation 4.1 and 4.2 together prove the lemma. ◀

▶ **Lemma 13.** *If $p$ does not divide the number of smallest degree factors of a monic square free $f(x) \in \mathbb{F}_q[x]$, then the smallest irreducible factor degree of $f(x)$ is $\deg(f(x)) - \deg(f(x) - \chi_f(x))$.*

**Proof.** Let $f(x) = \prod_i f_i(x)$ be a factorization of a monic square free $f(x) \in \mathbb{F}_q[x]$ into monic irreducible polynomials. Let $d$ be the smallest degree of factors of $f(x)$. Then

$$f(x) - \chi_f(x) = f(x) - \prod_i (f_i(x) - 1) = \sum_i \frac{f(x)}{f_i(x)} + (\text{terms of degree less than } \deg(f(x)) - d).$$

The first equality is from Lemma 12. Since $f(x)$ and $f_i(x)$ are all monic and $p$ does not divide the number of $f_i(x)$ of degree $d$, the leading term of $\sum_i (f(x)/f_i(x))$ is of degree $\deg(f(x)) - d$. Therefore $\deg(f(x) - \chi_f(x)) = \deg(f(x)) - d$ and the lemma follows. ◀

Lemma 13 reduces in nearly linear time Factor Degree (when restricted to $p$ greater than the number of factors of $f(x)$) to Carlitz Char-Poly. To see this, given $f(x)$, we may call an algorithm that solves Problem 11 to obtain $\chi_f(x)$ and output $\deg(f(x)) - \deg(f(x) - \chi_f(x))$.

# 5 Moore and Vandermonde Determinants

## 5.1 Moore Determinants and Carlitz Factorials

Let $A$ be a finitely generated $\mathbb{F}_q$ algebra and $n$ a positive integer. The Moore matrix $M_w$ with first row $w = (w_1, w_2, \ldots, w_n) \in A^n$ is defined as

$$M_w := \begin{bmatrix} w_1 & w_2 & w_3 & \ldots & w_n \\ w_1^q & w_2^q & w_3^q & \ldots & w_n^q \\ w_1^{q^2} & w_2^{q^2} & w_3^{q^2} & \ldots & w_n^{q^2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ w_1^{q^{n-1}} & w_2^{q^{n-1}} & w_3^{q^{n-1}} & \ldots & w_n^{q^{n-1}} \end{bmatrix}$$

and its determinant $det(M_w)$ is denoted by $\Delta(w_1, w_2, \ldots, w_n)$. For a positive integer $m$, the $m^{th}$ Carlitz factorial

$$\prod_{0 \le i < j \le m} \left(x^{q^{j-i}} - x\right)^{q^i},$$

is the product of all polynomials over $\mathbb{F}_q$ of degree at most $m$ [6]. We next recall Carlitz's identity and from it reduce Factor Degree to computing certain Moore determinants.

▶ **Lemma 14.** *(Carlitz [6]) For every positive integer $m$,*

$$\Delta(1, x, x^2, \ldots, x^m) = \prod_{0 \le i < j \le m} \left(x^{q^{j-i}} - x\right)^{q^i},$$

**Proof.** The Moore matrix with first row $(1, x, x^2, \ldots, x^m)$, when viewed column-wise is Vandermonde. By the Vandermonde determinant formula,

$$det\left(\begin{bmatrix} 1 & x & x^2 & \ldots & x^m \\ 1 & x^q & x^{2q} & \ldots & x^{mq} \\ 1 & x^{q^2} & x^{2q^2} & \ldots & x^{mq^2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x^{q^{n-1}} & x^{2q^{n-1}} & \ldots & x^{mq^{n-1}} \end{bmatrix}\right) = \prod_{0 \le i < j \le m} (x^{q^j} - x^{q^i}) = \prod_{0 \le i < j \le m} \left(x^{q^{j-i}} - x\right)^{q^i}$$

◀

Moore-Det may be restated as

▶ **Problem 15.** Given a square free monic polynomial $f(x) \in \mathbb{F}_q[x]$ of degree $n$ and a positive integer $m \le n$, decide if $\Delta(1, x, \ldots, x^m) \mod f(x)$ is 0.

Problem 15 can be solved in $\widetilde{O}(n^{3/2} \log q + n \log^2 q)$ time [13, Lemma 8.4].

▶ **Theorem 16.** *If there is a $T(n, m, \log q)$ time algorithm for Problem 15, then* Factor Degree *can be solved in $O(T(n, \lceil n/2 \rceil, \log q) \log n)$ time. That is,* Factor Degree *is nearly linear time reducible to* Moore-Det.

**Proof.** By Lemma 14, for a monic square free $f(x) \in \mathbb{F}_q[x]$ and $m \le \deg(f(x))$, we have $\Delta(1, x, \ldots, x^m) \mod f(x) = 0$ if and only if

$$\prod_{0 \le i < j \le m} \left(x^{q^{j-i}} - x\right)^{q^i} = 0 \mod f(x). \tag{5.1}$$

Since $f(x)$ is square free, Equation 5.1 holds if and only if every irreducible factor of $f(x)$ has degree at most $m$. Given oracle access to an algorithm for Problem 15, a binary search leads to the determination of the largest irreducible factor degree of $f(x)$. ◀

## 5.2 Vandermonde Determinants

The determinants involved in the previous subsection were both Moore and Vandermonde. Here we study determinants that are Vandermonde but not Moore. Further, the matrices involved are of dimension significantly smaller than the degree of the polynomial factored.

For a positive integer $m$, let

$$S_m := \{0, 1, 2, \ldots, \lfloor\sqrt{m}\rfloor - 1, \lfloor\sqrt{m}\rfloor, 2\lfloor\sqrt{m}\rfloor, 3\lfloor\sqrt{m}\rfloor, \ldots, (\lfloor\sqrt{m}\rfloor - 1)\lfloor\sqrt{m}\rfloor, \lfloor\sqrt{m}\rfloor^2, m\}.$$

This ensures that $|S_m| \leq 2\lfloor\sqrt{m}\rfloor + 1$ and $\{j - i | i, j \in S_m, i < j\} = \{1, 2, \ldots, m-1, m\}$.

For a positive integer $m$, let $V_m(x) \in \mathbb{F}_q[x]$ denote the determinant of the Vandermonde matrix with first row $\{x^{q^i}, i \in S_m\}$.

▶ **Lemma 17.** *For every monic square free $f(x) \in \mathbb{F}_q[x]$ and every positive integer $m$,*

$$\gcd(V_m(x), f(x)) = \gcd\left(\prod_{0 \leq i \leq m} \left(x^{q^i} - x\right), f(x)\right).$$

**Proof.** By the Vandermonde determinant formula,

$$V_m(x) = \prod_{i,j \in S_m | i<j} \left(x^{q^j} - x^{q^i}\right) = \prod_{i,j \in S_m | i<j} \left(x^{q^{j-i}} - x\right)^{q^i}. \tag{5.2}$$

Since $f(x)$ is square free and $\{j - i | i, j \in S_m, i < j\} = \{1, 2, \ldots, m-1, m\}$,

$$\gcd\left(\prod_{i,j \in S_m | i<j} \left(x^{q^{j-i}} - x\right)^{q^i}, f(x)\right) = \gcd\left(\prod_{0 \leq i \leq m} \left(x^{q^i} - x\right), f(x)\right). \tag{5.3}$$

By Equations 5.2 and 5.3, the lemma follows. ◀

Vandermonde Det may be restated as

▶ **Problem 18.** Given a square free monic polynomial $f(x) \in \mathbb{F}_q[x]$ of degree $n$ and a positive integer $m \leq n$, decide if $V_m(x) \mod f(x)$ is 0.

We next sketch a fast algorithm for Problem 18. Since $|S_m| \leq \sqrt{n}$, the first row $\{x^{q^i} \mod f(x), i \in S_m\}$ can be computed in $\widetilde{O}(n^{3/2} \log q + n \log^2 q)$ time using iterated Frobenius algorithm [21] implemented using fast modular composition [13]. Given the first row of a Vandermonde matrix over a commutative ring, the square of its determinant can be computed with nearly linearly many operations over the ring [16]. Hence, $V_m(x) \mod f(x)$ can be zero tested in $\widetilde{O}(n^{3/2} \log q + n \log^2 q)$ time.

▶ **Theorem 19.** *If there is a $T(n, m, \log q)$ time algorithm for Problem 18, then* Factor Degree *can be solved in $O(T(n, \lceil\sqrt{n}\rceil, \log q) \log n)$ time. That is,* Factor Degree *is nearly linear time reducible to* Vandermonde Det.

**Proof.** By Lemma 14 and Lemma 17, for every monic square free $f(x) \in \mathbb{F}_q[x]$ and positive integer $m \leq \deg(f(x))$,

$$\gcd(V_m(x), f(x)) = \gcd(\Delta(1, x, \ldots, x^m), f(x)).$$

Hence Problems 15 and 18 are identical and our theorem follows from Theorem 16. ◀

## References


1. A. Abboud, F. Grandoni, and V. V. Williams. Subcubic equivalences between graph centrality problems, APSP and diameter. In *Proceedings of the 26th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1681–1697, 2015.
2. A. Abboud and V. V. Williams. Popular conjectures imply strong lower bounds for dynamic problems. In *Proceedings of the 55th Annual Symposium on Foundations of Computer Science*, pages 434–443, 2014.
3. A. Abboud, V. V. Williams, and O. Weimann. Consequences of faster alignment of sequences. In *Automata, Languages, and Programming*, pages 39–51, 2014.
4. E. R. Berlekamp. Factoring polynomials over finite fields. *Bell System Technical Journal*, 46(8):1853–1859, 1967.
5. D. G. Cantor and H. Zassenhaus. A new algorithm for factoring polynomials over finite fields. *Mathematics of Computation*, 36(154):587–592, 1981.
6. L. Carlitz. On certain functions connected with polynomials in a Galois field. *Duke Math. J.*, 1(2):137–168, 06 1935.
7. L. Carlitz. A class of polynomials. *Transactions of the American Mathematical Society*, 43(2):167–182, 1938.
8. K. Conrad. Carlitz extensions, available online at. `http://www.math.uconn.edu/~kconrad/blurbs/gradnumthy/carlitz.pdf`.
9. P. Erdős. On the normal number of prime factors of $p-1$ and some related problems concerning Euler's $\phi$-function. *Quart. J. Math*, 6:205–213, 1935.
10. Z. Guo, A. K. Narayanan, and C. Umans. Algebraic problems equivalent to beating exponent 3/2 for polynomial factorization over finite fields. *arXiv preprint arXiv:1606.04592*, 2016.
11. E. Kaltofen and A. Lobo. Factoring high-degree polynomials by the black box berlekamp algorithm. In *Proceedings of the International Symposium on Symbolic and Algebraic Computation*, pages 90–98, 1994.
12. E. Kaltofen and V. Shoup. Subquadratic-time factoring of polynomials over finite fields. *Mathematics of computation*, 67(223):1179–1197, 1998.
13. K. S. Kedlaya and C. Umans. Fast polynomial factorization and modular composition. *SIAM J. Comput.*, 40(6):1767–1802, 2011.
14. D. E. Knuth. *The Art of Computer Programming, Volume 2 (3rd Ed.): Seminumerical Algorithms*. Addison-Wesley, 1997.
15. A. K. Narayanan. Polynomial factorization over finite fields by computing Euler-Poincare characteristics of Drinfeld modules. *arXiv preprint arXiv:1504.07697*, 2015.
16. V. Pan. On computations with dense structured matrices. *Mathematics of Computation*, 55(191):179–190, 1990.
17. M. Pătraşcu. Towards polynomial lower bounds for dynamic problems. In *Proceedings of the 42nd ACM Symposium on Theory of Computing*, pages 603–610, 2010.
18. L. Roditty and U. Zwick. Replacement paths and $k$ simple shortest paths in unweighted directed graphs. *ACM Trans. Algorithms*, 8(4):33:1–33:11, 2012.
19. J. B. Rosser and L. Schoenfeld. Approximate formulas for some functions of prime numbers. *Illinois J. Math.*, 6(1):64–94, 1962.
20. V. Shoup. Efficient computation of minimal polynomials in algebraic extensions of finite fields. In *Proceedings of the 1999 International Symposium on Symbolic and Algebraic Computation*, pages 53–58, 1999.
21. J. Von Zur Gathen and V. Shoup. Computing frobenius maps and factoring polynomials. *Computational complexity*, 2(3):187–224, 1992.
22. O. Weimann and R. Yuster. Replacement paths and distance sensitivity oracles via fast matrix multiplication. *ACM Trans. Algorithms*, 9(2):14:1–14:13, 2013.

**23** V. V. Williams. Faster replacement paths. In *Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1337–1346, 2011.

**24** V. V. Williams and R. Williams. Subcubic equivalences between path, matrix and triangle problems. In *Proceedings of the 51st Annual Symposium on Foundations of Computer Science*, pages 645–654, 2010.

**25** D. Y.Y. Yun. On square-free decomposition algorithms. In *Proceedings of the 3rd ACM Symposium on Symbolic and Algebraic Computation*, pages 26–35, 1976.