Using zeta functions to factor polynomials over finite fields

Bjorn Poonen

Abstract. In 2005, Kayal suggested that Schoof's algorithm for counting points on elliptic curves over finite fields might yield an approach to factor polynomials over finite fields in deterministic polynomial time. We present an exposition of his idea and then explain details of a generalization involving Pila's algorithm for abelian varieties.

1. Introduction

Factoring univariate polynomials over finite fields is a solved problem in practice. Known algorithms are fast, and they are proved to run in polynomial time if granted access to a source of randomness. But the theoretical question of whether there exists a *deterministic* polynomial-time algorithm remains open. See the survey articles [**Len82**], [**Len90**], and [**vzGP01**]; the last of these contains a very extensive bibliography.

In 1985, Schoof gave a deterministic polynomial-time algorithm to compute the number of points on a given elliptic curve over a finite field [**Sch85**, Section 3]. At the Mathematisches Forschungsinstitut Oberwolfach in July 2005, Neeraj Kayal suggested a way to use Schoof's algorithm to attempt to factor polynomials over finite fields in deterministic polynomial time. The author, who was present, responded that one could use higher genus curves or higher-dimensional abelian varieties in place of elliptic curves, and that these heuristically had a greater chance of success.

It seems that the only written record of the ideas of Kayal and the author before now is the 2006 master's thesis of Amalaswintha Wolfsdorf [**Wol06**]. She describes Kayal's idea for elliptic curves in detail, and writes a few sentences on the higher genus case based on a November 25, 2005 email from the present author. Our purpose is to present a brief exposition of Kayal's idea and to explain details of the generalization, which is Theorem 5.1 in this article.

2. Schoof's algorithm

Understanding Kayal's idea requires some knowledge of Schoof's algorithm, which we now recall, in the special case of a prime field $\mathbb{F}_p$.

2010 *Mathematics Subject Classification.* Primary 11Y05; Secondary 14G10, 14G15, 14K15.

Key words and phrases. Factoring polynomials, zeta function, abelian variety, finite field.

This research was supported in part by National Science Foundation grant DMS-1601946 and Simons Foundation grants #402472 (to Bjorn Poonen) and #550033.

Theorem 2.1 ([**Sch85**, Section 3]). *There exists a deterministic polynomial-time algorithm that takes as input a prime p and a Weierstrass equation of an elliptic curve E over $\mathbb{F}_p$, and outputs $\#E(\mathbb{F}_p)$.*

Polynomial-time means polynomial in the size of the input, which is of order $\log p$.

Sketch of proof. Hasse proved that $\#E(\mathbb{F}_p) = p - a + 1$ for some $a \in \mathbb{Z}$ satisfying $|a| \le 2\sqrt{p}$. If one can compute $a \bmod \ell$ for all primes $\ell \ne p$ up to some bound L, then an effective Chinese remainder theorem lets one compute $a \bmod \prod_{\ell<L} \ell$. If L is chosen as a sufficiently large constant multiple of $\log p$, then $\prod_{\ell<L} \ell > 4\sqrt{p}$, so $a \bmod \prod_{\ell<L} \ell$ determines a.

The Frobenius endomorphism F of E satisfies $F^2 - aF + p = 0$ in $\operatorname{End} E$. In particular, $F^2 - aF + p$ acts as 0 on the ℓ-torsion subscheme $E[\ell]$, and this condition uniquely determines $a \bmod \ell$. It remains to explain how to compute with these objects. First, $E[\ell]$ is $\operatorname{Spec} R$ for some $\mathbb{F}_p$-algebra R defined by $O(1)$ explicit equations of degree polynomial in ℓ, and these equations can be computed from the group law on E; from this, one can compute an explicit multiplication table for R with respect to an $\mathbb{F}_p$-basis. (With a little more work, following Schoof, one can work even more explicitly by using division polynomials, but this does not generalize as easily.) The action of F on $E[\ell]$ is given by the pth power map on R, whose action on $\mathbb{F}_p$-algebra generators can be computed explicitly by writing the exponent p in binary and using repeated squaring and multiplication. Similarly, the action of p (or any smaller integer) on $E[\ell]$ can be computed by writing p in binary and using repeated doubling and addition on E. Combining these lets one compute the action of $F^2 - aF + p$ on $E[\ell]$ in time bounded by $P(\ell, \log p)$ for some polynomial P. For each ℓ, try $a = 0, 1, \ldots, \ell - 1$ until the value mod ℓ is found that makes $F^2 - aF + p$ kill $E[\ell]$. The total running time is at most $\sum_{\ell<L} \sum_{a=0}^{\ell-1} P(\ell, \log p)$, which is polynomial in $\log p$. □

3. Kayal's factoring idea

For simplicity, suppose that p is a large prime and suppose that we are given the product $f(t) = (t - r_1)(t - r_2) \in \mathbb{F}_p[t]$ for some unknown distinct $r_1, r_2 \in \mathbb{F}_p$. Let $B = \mathbb{F}_p[t]/(f(t))$, so B is secretly isomorphic to $\mathbb{F}_p \times \mathbb{F}_p$. Elements of B are represented by polynomials of degree ≤ 1 in $\mathbb{F}_p[t]$. A Weierstrass equation over B with discriminant in $B^\times$ defines an elliptic scheme E over B, which secretly specializes to two elliptic curves over $\mathbb{F}_p$, say E_1 and E_2.

What happens if we blithely run Schoof's algorithm on E, as if B were $\mathbb{F}_p$? If $\#E_1(\mathbb{F}_p) = \#E_2(\mathbb{F}_p)$, then there exists $a \in \mathbb{Z}$ such that $F^2 - aF + p = 0$ in $\operatorname{End} E$, and the algorithm runs as usual, and outputs the common value $\#E_1(\mathbb{F}_p) = E_2(\mathbb{F}_p)$, but we learn nothing about the factorization of $f(t)$. Now suppose instead that $\#E_1(\mathbb{F}_p) \ne \#E_2(\mathbb{F}_p)$. Write $\#E_i(\mathbb{F}_p) = p - a_i + 1$ for $i = 1, 2$, so $a_1 \ne a_2$. Then for some ℓ, we have $a_1 \not\equiv a_2 \pmod{\ell}$. Thus, when we check integers a to see if $F^2 - aF + p$ kills $E[\ell]$, which amounts to certain elements of B vanishing, we instead find an integer a_1 for which these elements of B vanish mod $t - r_1$ but do not all vanish mod $t - r_2$. Hence we discover a nontrivial factor of $f(t)$.

Heuristically it is likely that $\#E_1(\mathbb{F}_p) \ne \#E_2(\mathbb{F}_p)$, since there are about $4\sqrt{p}$ possible values for the order of an elliptic curve over $\mathbb{F}_p$. If we are unlucky enough to have chosen E so that $\#E_1(\mathbb{F}_p) = E_2(\mathbb{F}_p)$, we can try again with a different E, or use the same linear polynomials as Weierstrass coefficients while replacing $f(t)$ by $f(t+1)$. We do not have a proof, however, that a deterministic sequence of such trials will succeed after polynomially many attempts.

REMARK 3.1. The same approach can be tried to factor a polynomial $f(t) := (t-r_1)\cdots(t-r_d)$ for distinct $r_1, \ldots, r_d \in \mathbb{F}_p$, by induction on d. If, using obvious notation, $\#E_1(\mathbb{F}_p), \ldots, \#E_d(\mathbb{F}_p)$ are not all equal, then Schoof's algorithm will find a nontrivial factor g of f, and then we can apply the inductive hypothesis to factor g and f/g.

REMARK 3.2. Berlekamp [**Ber70**] showed that one can reduce the problem of factoring polynomials in $\mathbb{F}_q[t]$ for arbitrary prime powers q to the problem of factoring polynomials in $\mathbb{F}_p[t]$ with distinct roots all in $\mathbb{F}_p$; see also [**Len82**, Sections 3 and 4] for another exposition of this.

4. Pila's algorithm

To generalize Kayal's approach to abelian varieties, we need Pila's generalization of Schoof's algorithm.

Let A be a g-dimensional abelian variety over $\mathbb{F}_p$. Let F be the Frobenius endomorphism of A. For each prime $\ell \neq p$, we may form the ℓ-adic Tate module $T_\ell A := \varprojlim_n A[\ell^n]$. Let $P(t)$ be the characteristic polynomial of F acting on $T_\ell A$, so $\deg P = 2g$. A priori the coefficients of P are in $\mathbb{Z}_\ell$, but in fact they lie in $\mathbb{Z}$ and are independent of the choice of ℓ. Knowledge of P is equivalent to knowledge of the zeta function Z_A.

An abelian variety A over $\mathbb{F}_p$ can be described explicitly by giving a positive integer N and a finite list of homogeneous polynomials in $\mathbb{F}_p[x_0, \ldots, x_N]$ whose common zero locus in $\mathbb{P}^N$ is A, together with the addition morphism $A \times A \to A$ (and also the inversion morphism if desired) in terms of explicit polynomial mappings on affine patches. Pila's algorithm accepts such data as input, and outputs $P(t) \in \mathbb{Z}[t]$. Its running time is bounded by a polynomial in $\log p$ whose degree and coefficients depend only on N and the number and degrees of the polynomials defining A and the addition law [**Pil90**, Theorem A].

The general outline of Pila's algorithm is similar to that of Schoof's algorithm: it computes $P(t) \bmod \ell$ for many small primes ℓ by studying the action of F on $A[\ell]$, and then reconstructs $P(t)$ by using an effective Chinese remainder theorem. For each ℓ, it tries each monic degree $2g$ polynomial in $(\mathbb{Z}/\ell\mathbb{Z})[t]$ and tests whether it equals $P(t) \bmod \ell$. Each test involves a deterministic sequence of arithmetic operations on elements of $\mathbb{F}_p$ controlled by queries: each query asks whether some previously computed element is 0, and the result dictates which arithmetic operation is to be carried out next. This is all that we will need to know about Pila's algorithm.

5. Generalization of Kayal's factoring idea

We will prove that we can replace elliptic curves by abelian varieties in Kayal's approach. Also, instead of using only the order of the group of points, we can use the whole zeta function. The advantage of using higher-dimensional abelian varieties is that there are many more possible zeta functions, so success becomes very likely, at least heuristically: see Section 6.

Given a variety V over a finite field, let Z_V be its zeta function, viewed as a rational function in $\mathbb{Q}(T)$.

Let U be a dense open subscheme of $\mathbb{A}^1_{\mathbb{Z}} := \operatorname{Spec} \mathbb{Z}[t]$. Let $\mathcal{A} \to U$ be an abelian scheme. For each prime p and $u \in U(\mathbb{F}_p)$, let $\mathcal{A}_u$ be the fiber above u. Concretely, $\mathcal{A}$ can be thought of as a family of abelian varieties defined by equations with coefficients in $\mathbb{Z}[t]$; specializing t to a suitably general value $u \in \mathbb{F}_p$ produces an abelian variety $\mathcal{A}_u$ over $\mathbb{F}_p$; here "suitably general" means outside a certain bad locus, which may be taken to be of the form $\Delta(t) = 0$ for some "discriminant" $\Delta(t) \in \mathbb{Z}[t]$ that is not identically zero but vanishes at any u for which the specialization $\mathcal{A}_u$ is degenerate.

Theorem 5.1 below will involve the following:

Hypothesis Z. There exist a dense open subscheme U of $\mathbb{A}^1_{\mathbb{Z}}$ and an abelian scheme $\mathcal{A} \to U$ such that for every sufficiently large prime p, the $Z_{\mathcal{A}_u}$ for the different $u \in U(\mathbb{F}_p)$ are distinct.

Theorem 5.1. *There is a deterministic algorithm that takes as input a finite field $\mathbb{F}_q$ and a nonzero polynomial $f \in \mathbb{F}_q[t]$, and outputs the factors of f in $\mathbb{F}_q[t]$, such that if Hypothesis Z holds, then the running time is polynomial in* $\log q$ *and* $\deg f$.

Remark 5.2. Our proof will show that an algorithm as in Theorem 5.1 not only exists, but also can be written down explicitly, even if we do not know in advance the abelian scheme $\mathcal{A} \to U$ in Hypothesis Z.

Proof. By Remark 3.2, we may assume that q is a prime p and that f has distinct roots all in $\mathbb{F}_p$. Of course we also assume that $\deg f \geq 2$.

First, we give an algorithm depending on explicit knowledge of an abelian scheme $\mathcal{A} \to U$ as in Hypothesis Z. More precisely, we may shrink U to assume that $U = \operatorname{Spec} T$, where $T = \mathbb{Z}[t][1/\Delta]$ for some nonzero polynomial $\Delta \in \mathbb{Z}[t]$, and we may assume that we are given explicit polynomials describing $\mathcal{A}$ over T in the same way that we described abelian varieties over $\mathbb{F}_p$ in Section 4.

There are at most $(\deg f)(\deg \Delta)$ values $c \in \mathbb{F}_p$ such that $f(t+c)$ and $\Delta(t)$ have a nontrivial gcd, so by trying $c = 0, 1, \ldots$ in turn, we quickly find such a c (of course, we may assume that $p > (\deg f)(\deg \Delta)$. Replace $f(t)$ by $f(t+c)$ to assume that $\gcd(f, \Delta) = 1$. Let $B = \mathbb{F}_p[t]/(f(t))$. Then $\operatorname{Spec} B$ is a closed subscheme of U, and the base change $\mathcal{A}_B$ is an abelian scheme over B. It consists of a disjoint union of abelian varieties $\mathcal{A}_u$ over $\mathbb{F}_p$, one for each zero u of f.

Apply Pila's algorithm to $\mathcal{A}_B$, but each time it queries an element of B to test whether it is 0, instead compute a gcd with f to test whether it is 0, a unit, or a nonzero zerodivisor. By Hypothesis Z, the zeta functions $Z_{\mathcal{A}_u}$ for two different zeros u of f are distinct in $\mathbb{Q}(T)$, so there exists a prime ℓ such that the characteristic polynomials mod ℓ of $\mathcal{A}_u$ for these two u are distinct. Therefore the computations in Pila's algorithm must eventually diverge for these two values of u, which can happen only if a nonzero zerodivisor in B is encountered. At that point, we have found a nontrivial factor f_0 of f. Apply induction to the factors f_0 and f/f_0. This completes the description of the algorithm when we are given $\mathcal{A} \to U$ explicitly. In particular, there exists an algorithm to factor polynomials, even though we might not know which algorithm it is that does it.

We now describe a new program Ω that does not rely on knowledge of $\mathcal{A} \to U$. Program Ω enumerates all computer programs and runs them in parallel, devoting a fraction 2^{-n} of its computing power to the nth program; at each step of each program, Ω tests whether what that program has printed so far is a list of linear polynomials over $\mathbb{F}_p$ whose product is f, and if so, Ω terminates the whole computation with this answer. If Hypothesis Z is true, and n is the number of the program described in earlier paragraphs using an abelian scheme $\mathcal{A} \to U$ as in Hypothesis Z, then Ω finds the factorization in time bounded by 2^n times a polynomial, but 2^n is a constant, so this is still polynomial in the size of the input. If Hypothesis Z is false, then Ω still terminates with the correct factorization because there exists N such that program N factors polynomials by trial division, but the running time of Ω is not guaranteed to be bounded by a polynomial in this case. □

6. A heuristic for Hypothesis Z

For a g-dimensional abelian variety A over $\mathbb{F}_p$, the complex zeros of the characteristic polynomial $P(t)$ have absolute value $p^{1/2}$, so the coefficient of t^{2g-m} in $P(t)$ is $O_g(p^{m/2})$, with the implied constant depending on g but not p. Also, the functional equation of Z_A shows that the coefficient of t^m in $P(t)$ is determined by the coefficient of t^{2g-m}. Thus $P(t)$ is determined by coefficients of t^{2g-m} for $m = 1, 2, \ldots, g$, so there are at most $\prod_{m=1}^{g} O_g(p^{m/2}) = O_g(p^{g(g+1)/4})$ possibilities for $P(t)$. Equivalently, if $\mathcal{Z}_{g,p}$ is the set of zeta functions of all g-dimensional abelian varieties over $\mathbb{F}_p$, then $\#\mathcal{Z}_{g,p} = O_g(p^{g(g+1)/4})$ as $p \to \infty$. In fact, DiPippo and Howe [**DH98**] prove that for fixed g, we have $\#\mathcal{Z}_{g,p} \sim p^{g(g+1)/4}$ as $p \to \infty$, where in this section we use the notation $f(p) \sim h(p)$ to mean that $f(p)/h(p)$ tends to a positive constant depending only on g as $p \to \infty$.

If we sample about p zeta functions from $\mathcal{Z}_{g,p}$ at random, then the expected number of equal pairs is $\sim \binom{p}{2} \frac{1}{p^{g(g+1)/4}} \sim p^{2-g(g+1)/4}$. If we do this for all primes p greater than or equal to some large integer p_0, then the expected total number of equal pairs for all p is $\sum_{p \ge p_0} p^{2-g(g+1)/4}$, which tends to 0 as $p_0 \to \infty$, provided that $2 - g(g+1)/4 < -1$, which holds for $g \ge 4$.

Now let U be a dense open subscheme of $\mathbb{A}^1_{\mathbb{Z}}$, and let $\mathcal{A} \to U$ be an abelian scheme of relative dimension g. The previous paragraph suggests that if we model the zeta functions of the fibers of $\mathcal{A} \to U$ above $\mathbb{F}_p$-points of U as being independent random elements of $\mathcal{Z}_{g,p}$, then for sufficiently large p_0 it is true for every $p \ge p_0$ that these zeta functions will be distinct; in other words, $\mathcal{A} \to U$ should satisfy the condition in Hypothesis Z, unless there is some extra structure to the family that the model fails to reflect.

It even seems reasonable to guess that for a typical 1-parameter family of genus 4 curves, the family of Jacobians will satisfy the condition in Hypothesis Z. See [**SV17**] for some specific candidate families.

7. Weakening Hypothesis Z

Something slightly weaker than Hypothesis Z would suffice to obtain a polynomial running time in Theorem 5.1:

HYPOTHESIS Z′. There exist a dense open subscheme U of $\mathbb{A}^1_{\mathbb{Z}}$ and an abelian scheme $\mathcal{A} \to U$ such that for every sufficiently large prime p, there are at least $p - (\log p)^{O(1)}$ distinct zeta functions $Z_{\mathcal{A}_u}$ as u varies over $U(\mathbb{F}_p)$.

Under Hypothesis Z′, given $f \in \mathbb{F}_p[t]$ that factors completely, one can attempt to factor the polynomials $f(t+c)$ for $(\log p)^{O(1)}$ different values $c \in \mathbb{F}_p$ by running

Pila's algorithm as in the proof of Theorem 5.1. As long as the $O(1)$ here is larger than the $O(1)$ in Hypothesis Z$'$, and as long as p is sufficiently large, there will be at least one such c such that all the zeros of $f(t+c) \bmod p$ lie in $U(\mathbb{F}_p)$ and the zeta functions of the fibers above these zeros are pairwise distinct. Thus the algorithm will succeed in factoring $f(t+c)$ for at least one c, and evaluating the factors at $t-c$ recovers the factorization of $f(t)$. The running time of the algorithm is still polynomial in $\log p$, albeit possibly with a larger exponent.

8. Using varieties other than abelian varieties

Suppose that instead of an abelian scheme as in Hypothesis Z, one had an arbitrary finite-type morphism $\mathcal{X} \to U$ for a dense open subscheme U of $\mathbb{A}^1_{\mathbb{Z}}$ such that for any sufficiently large prime p and distinct $u_1, u_2 \in U(\mathbb{F}_p)$, there exists a prime ℓ bounded by a polynomial in $\log p$ and a nonnegative integer i such that the characteristic polynomials of Frobenius acting on $\mathrm{H}^i_{\mathrm{et}}(\mathcal{X}_{u_1} \times \overline{\mathbb{F}}_p, \mathbb{Z}/\ell\mathbb{Z})$ and $\mathrm{H}^i_{\mathrm{et}}(\mathcal{X}_{u_2} \times \overline{\mathbb{F}}_p, \mathbb{Z}/\ell\mathbb{Z})$ are different. Then again one could factor polynomials over finite fields in deterministic polynomial time, provided that one had an analogue of Pila's algorithm that could compute these characteristic polynomials using a deterministic sequence of arithmetic operations and queries whose number is bounded by a polynomial in ℓ whose degree and coefficients depend only on $\mathcal{X} \to U$.

Madore and Orgogozo [**MO15**, Théorème 0.1] gave an algorithm for computing such characteristic polynomials, but their bound on the running time is only primitive recursive, not polynomial in ℓ.

Acknowledgements

I thank Andrew Sutherland and José Felipe Voloch for encouraging me to write this article, and I thank Amalaswintha Wolfsdorf for sharing her master's thesis with me.

References

[Ber70] E. R. Berlekamp, *Factoring polynomials over large finite fields*, Math. Comp. **24** (1970), 713–735, DOI 10.2307/2004849. MR0276200 ↑3

[DH98] Stephen A. DiPippo and Everett W. Howe, *Real polynomials with all roots on the unit circle and abelian varieties over finite fields*, J. Number Theory **73** (1998), no. 2, 426–450, DOI 10.1006/jnth.1998.2302. Corrigendum in J. Number Theory **83** (2000), no. 1, 182. MR1657992 ↑5

[Len82] A. K. Lenstra, *Factorization of polynomials*, Computational methods in number theory, Part I, Math. Centre Tracts, vol. 154, Math. Centrum, Amsterdam, 1982, pp. 169–198. MR700263 ↑1, 3

[Len90] H. W. Lenstra Jr., *Algorithms for finite fields*, Number theory and cryptography (Sydney, 1989), London Math. Soc. Lecture Note Ser., vol. 154, Cambridge Univ. Press, Cambridge, 1990, pp. 76–85. MR1055400 ↑1

[MO15] David A. Madore and Fabrice Orgogozo, *Calculabilité de la cohomologie étale modulo ℓ*, Algebra Number Theory **9** (2015), no. 7, 1647–1739, DOI 10.2140/ant.2015.9.1647 (French, with English and French summaries). MR3404650 ↑6

[Pil90] J. Pila, *Frobenius maps of abelian varieties and finding roots of unity in finite fields*, Math. Comp. **55** (1990), no. 192, 745–763, DOI 10.2307/2008445. MR1035941 ↑3

[Sch85] René Schoof, *Elliptic curves over finite fields and the computation of square roots mod p*, Math. Comp. **44** (1985), no. 170, 483–494, DOI 10.2307/2007968. MR777280 ↑1, 2

[SV17] Andrew V. Sutherland and José Felipe Voloch, *Maps between curves and arithmetic obstructions*, September 18, 2017. Preprint, `arXiv:1709.05734v1`. ↑5 [vzGP01] Joachim von zur Gathen and Daniel Panario, *Factoring polynomials over finite fields: a survey*, J. Symbolic Comput. **31** (2001), no. 1-2, 3–17, DOI 10.1006/jsco.1999.1002. Computational algebra and number theory (Milwaukee, WI, 1996). MR1806203 ↑1

[Wol06] Amalaswintha Wolfsdorf, *Factorising polynomials over finite fields*, March 2006. Master's thesis, University of Oxford, `arXiv:1709.05513v1`. ↑1

Department of Mathematics, Massachusetts Institute of Technology, Cambridge, MA 02139-4307, USA

E-mail address: `poonen@math.mit.edu`

URL: `http://math.mit.edu/~poonen/`