Factoring Polynomials over Finite Fields by Modular Polynomial Composition

Erich Kaltofen

North Carolina State University
Departments of Mathematics and Computer Science
Raleigh, North Carolina, USA
Email: kaltofen@eos.ncsu.edu
URL: http://www4.ncsu.edu/~kaltofen

Joint work with Victor Shoup

Factorization of an integer N
(quadratic sieves, number field sieves)

Compute a solution to the congruence equation

$$X^2 \equiv Y^2 \pmod{N}$$

via r relations on b basis primes

$$X_1^2 \cdot X_2^2 \cdots X_r^2 \equiv (p_1^{e_1})^2 \cdot (p_2^{e_2})^2 \cdots (p_b^{e_b})^2 \pmod{N}$$

Then N divides $(X+Y)(X-Y)$, hence

$$\mathrm{GCD}(X+Y, N) \text{ divides } N$$

Factorization of polynomial f over finite field $\mathbb{F}_p$
(Berlekamp 1967 algorithm)

Note that since $a^p \equiv a \pmod{p}$ for all $a \in \mathbb{F}_p$ we have

$$x^p - x \equiv x \cdot (x-1) \cdot (x-2) \cdots (x-p+1) \pmod{p}$$

Compute a polynomial solution to the congruence equation

$$w(x)^p \equiv w(x) \pmod{f(x)}$$

Then f divides $w \cdot (w-1) \cdot (w-2) \cdots (w-p+1)$, hence

$$\mathrm{GCD}(w(x) - a, f(x)) \text{ divides } f(x) \text{ for some } a \in \mathbb{F}_p$$

Solving $w^p \equiv w \pmod{f}$ by linear algebra

For $w(x) = w_0 + w_1 x + \cdots + w_{n-1}x^{n-1} \in \mathbb{F}_p[x]$, $n = \deg(f)$:

$$w(x)^p = w(x^p) \equiv w(x) \pmod{f(x)}$$

(Note: $(a+b)^p = a^p + b^p$ because $\binom{p}{i} \equiv 0 \pmod{p}$ for $0 < i < p$)

$$\Updownarrow$$

$$\overrightarrow{w(x^p) \bmod f(x)}^{\mathrm{tr}} = \underbrace{[\, w_0 \ldots w_{n-1} \,]}_{\overrightarrow{w}^{\mathrm{tr}}} \cdot \underbrace{\begin{bmatrix} \vdots \\ \overrightarrow{x^{ip} \bmod f(x)}^{\mathrm{tr}} \\ \vdots \end{bmatrix}_{0 \le i < n}}_{Q} = \overrightarrow{w}^{\mathrm{tr}}$$

(Petr's 1937 matrix)

Run-time comparisons (field arithmetic operations)

	$q = O(1)$	$\log q = \Theta(n)$
Berlekamp '70 $O(n^{\omega} + n^{1+o(1)} \log q)$	$O(n^{2.38})$	$O(n^{2.38})$
Cantor & Zassenhaus '81 $O(n^{2+o(1)} \log q)$	$O(n^{2+o(1)})$	$O(n^{3+o(1)})$
von zur Gathen & Shoup '91 $O(n^{2+o(1)} + n^{1+o(1)} \log q)$	$O(n^{2+o(1)})$	$O(n^{2+o(1)})$
Kaltofen & Shoup '94 $O(n^{(\omega+1)/2+(1-\gamma)(\omega-1)/2} + n^{1+\gamma+o(1)} \log q)$ for any $0 \leq \gamma \leq 1$	$O(n^{1.82})$	$O(n^{2.5})$

ω = matrix multiplication exponent

The high algebraic extension case (May 9, 1996)

Let $q = 2^{\lceil n^{1.5} \rceil}$, and consider factorization over $\mathbb{F}_q$ in terms of bit complexity.

von zur Gathen & Shoup '91: $O(n(\log q)^2)$.

Kaltofen & Shoup '96: $O(n(\log q)^{1.69})$.

Generalizes to $q = p^k$ where k grows superlinearly with n.

Distinct degree factorization (Arwin 1918)

Fact: $x^{q^i} - x = \prod_{\substack{f \text{ irreducible over } \mathbb{F}_q \\ \deg(f) \text{ divides } i}} f(x)$

Write $f^{[i]} = \prod_{\substack{g \text{ irred. factor of } f \\ \deg(g) = i}} g$

$f^* \leftarrow f$; `/* squarefree */`

`for` $i \leftarrow 1, \ldots, \lfloor n/2 \rfloor$ `do`

`{`$f^{[i]}(x) \leftarrow \mathrm{GCD}(-x + x^{q^i} \bmod f^*(x), f^*(x))$;

$f^* \leftarrow f^* \Big/ f^{[i]}$;

`}`

$f^{[\deg(f^*)]} \leftarrow f^*$; `/* factor with degree` $> \lfloor n/2 \rfloor$ `*/`

Suppose $f(x) \in \mathbb{F}_q[x]$ has degree n, $g(x)$, $h(x)$ are modular residues.
All counts are in terms of arithmetic operations in $\mathbb{F}_q$.

Problem	Complexity	Inventors of algorithm
1. $g \cdot h \pmod f$	$O(n(\log n)\operatorname{loglog} n)$	Schönhage&Strassen 1969 Schönhage 1977 ($p = 2$)
2. $\operatorname{GCD}(f, g)$	$O(n(\log n)^2 \operatorname{loglog} n)$	Knuth 1971/Moenck 1973
3. $g^q \pmod f$	$O((\log q)n^{1+o(1)})$	Pingala 200 b.c.
4. $g(h(x)) \pmod{f(x)}$	$O(n^{1.69})$	Brent&Kung 1978 Coppersmith&Winograd 1987
5. $x^{q^n} \pmod{f(x)}$ given $x^q \pmod{f(x)}$	$O(n^{1.69})$	von zur Gathen&Shoup 1991

6. $g(h_1), ..., g(h_n) \pmod{f}$ $O(n^{2+o(1)})$ Moenck&Borodin 1972

7. $x^{q^2}, ..., x^{q^n} \pmod{f(x)}$ $O(n^{2+o(1)})$ von zur Gathen&Shoup 1991
   given $x^q \pmod{f(x)}$

Fast computation of $x^{q^n} \bmod f(x)$

$$x^{q^i} \equiv (\underbrace{x^{q^{i-1}}}_{h_{i-1}(x)})^q$$

$$\equiv h_{i-1}(\underbrace{x^q}_{h_1(x)})$$

$$\equiv h_{i-1}(\, h_1(x)\,)$$

$$\equiv h_{\lfloor i/2 \rfloor}(\, h_{\lfloor i/2 \rfloor}(\, h_{i \bmod 2}(x)\,)\,) \pmod{f(x)}$$

(modular polynomial composition)

Fast modular polynomial composition

Compute $g(h(x)) \pmod{f(x)}$ with $O(n^{1.69})$ field operations.

$$g(x) = \sum_{j=0}^{\lceil\sqrt{n}\rceil} \left(\sum_{l=0}^{\lfloor\sqrt{n}\rfloor - 1} c_{j,l} x^l \right) \cdot x^{\lfloor\sqrt{n}\rfloor \cdot j}$$

$$[c_{j,l}] \quad \cdot \quad \begin{bmatrix} \overrightarrow{h^0 \bmod f} \\ \overrightarrow{h^1 \bmod f} \\ \overrightarrow{h^2 \bmod f} \\ \vdots \\ \overrightarrow{h^{\lfloor\sqrt{n}\rfloor - 1} \bmod f} \end{bmatrix}$$

$\lfloor\sqrt{n}\rfloor \times \lfloor\sqrt{n}\rfloor \quad \lfloor\sqrt{n}\rfloor \times n \qquad\qquad \Rightarrow O(\sqrt{n}(\sqrt{n})^{2.38})$

Baby step-giant step algorithm (K and Shoup 1994)

Fact: $x^{q^J} - x^{q^i} = \left(x^{q^{J-i}} - x\right)^{q^i} = \left(\prod_{\substack{f \text{ irreducible over } \mathbb{F}_q \\ \deg(f) \text{ divides } J-i}} f(x) \right)^{q^i}$

Let $l = \lceil n^\beta \rceil$ with $0 \leq \beta \leq 1$:

$$\mathrm{GCD}\Big(\prod_{i=0}^{l-1} (\underbrace{x^{q^{jl}}}_{\| H_j} - \underbrace{x^{q^i}}_{\| h_i}) \bmod f(x), \;\; f(x) \Big)$$

has all those factors of f whose degree is in the interval $[(j-1)l+1, jl]$.

Step 1 (*baby steps*): Let $l = \lceil n^{\beta} \rceil$.
`for` $i \leftarrow 1, \ldots, l-1$ `do` $h_i(x) \leftarrow x^{q^i} \bmod f(x)$.

Cost: $O(n^{1+\beta+o(1)} \log q)$

Step 2 (*giant steps*):
`for` $j \leftarrow 1, \ldots, \lceil n/(2l) \rceil$ `do` $H_j(x) \leftarrow x^{q^{jl}} \bmod f(x)$.

Cost: $O(n^{1.69+(1-\beta)})$

Step 3 (*coarse distinct degree factorization*):

`for` $j \leftarrow 1, \ldots, \lceil n/(2l) \rceil$ `do` $I_j \leftarrow \prod_{i=0}^{l-1} (H_j - h_i) \bmod f$.

$f^* \leftarrow f$;
`for` $j \leftarrow 1, \ldots, \lceil n/(2l) \rceil$ `do`
$\{F_j \leftarrow \mathrm{GCD}(I_j, f^*); f^* \leftarrow f^* \Big/ F_j\}$

Step 4 (*fine distinct degree factorization*):
`Split` $F_j = f^{[(j-1)l+1]} \cdot f^{[(j-1)l+2]} \ldots f^{[jl]}$ `á la Arwin.`

Lobo's '94 implementation of black box Berlekamp
(based on block Wiedemann)

Degree n	Prime p	Task	# Computers 8	32	Factor degrees
15001	127	Step W1	$82^h20'$		1, 1, 2, 2, 4, 12
		Step W2	$12^h53'$		21, 21, 33, 55
		Step W3	$42^h42'$		155, 158, 351
		split/refine	$3^h19'$		809, 1793, 2665
		total time	$141^h14'$		2813, 2919, 3186
		work	$87577^{\#}$		

Parallel CPU time (hours^{h}minutes$'$) for factoring

$$(x^{7501} + x + 1) \cdot (x^{7500} + x + 1) \pmod{127}$$

on 86.1 MIPS computers; work is measured in MIPS-hours$^{\#}$

Shoup's baby step/giant step implementation

Can factor a 2048 degree pseudo-random polynomial modulo a 2048 bit prime number in about 12 days on a single Sparc-10 computer.

The algorithm requires 68 Mbytes of memory.

Note: Shoup implemented a variant based on the distinct-degree factorization algorithm

von zur Gathen and Gerhard's implementation over $\mathbb{F}_2$ (ISSAC '96)

Can factor a 262143 degree pseudo-random polynomial modulo 2 in about 48 CPU hours using 2 Ultrasparc 1 computers.

The algorithm requires 1 Gbytes of hard disk space.

Note: von zur Gathen and Gerhard implemented special purpose polynomial arithmetic over $\mathbb{F}_2$ due to Cantor '89 and used the distinct-degree factorization approach.

Suppose $f = f_1 \cdots f_r$ where $\deg(f_1) = \deg(f_2) = \cdots = \deg(f_r) = d$. Then

$$v(x) + v(x)^q + v(x)^{q^2} + \cdots + v(x)^{q^{d-1}} \bmod f_i(x) \quad \in \quad \mathbb{F}_q$$

"trace of Frobenius"

One can compute

$$v(x)^{q^i} \equiv v(\underbrace{x^{q^i}}_{h_i(x) \equiv h_{i-1}(h_1(x)) \pmod{f(x)}}) \equiv v(h_i(x)) \pmod{f(x)} \quad \text{in } O(n^{1.69+o(1)})\ \mathbb{F}_q\text{-ops (given } h_1, h_{i-1})$$

More efficiently (von zur Gathen/Shoup '92), one "doubles"

$$x^{q^{2i}} \equiv (x^{q^i})^{q^i} \equiv h_i(x)^{q^i} \equiv h_i(x^{q^i}) \equiv h_i(h_i(x)) \pmod{f(x)}$$

and

$$(\underbrace{v(x)^q + v(x)^{q^2} + \cdots + v(x)^{q^i}}_{w_i(x)})^{q^i} \equiv \begin{cases} w_i(x)^{q^i} \equiv w_i(h_i) \\ v(x)^{q^{i+1}} + \cdots + v(x)^{q^{2i}} \equiv w_{2i}(x) - w_i(x) \end{cases}$$

hence finds the entire trace of Frobenius in $O(n^{1.69})$ $\mathbb{F}_q$-ops (given h_1).

Computing $x^q \bmod f(x)$ with $f(x) \in \mathbb{F}_q[x]$ where $q = p^k$ even faster

Suppose $\mathbb{F}_q = \mathbb{F}_p[z]/(\varphi(z))$ and we already have

$$x^{p^i} \bmod f(x) = h_i(x) = c_0(z) + c_1(z)x + \cdots + c_{n-1}(z)x^{n-1} \in \mathbb{F}_q[x].$$

and

$$z^{p^i} \bmod \varphi(z) = \psi(z) \in \mathbb{F}_p[z].$$

Then

$$\begin{aligned}
x^{p^{2i}} &\equiv (c_0(z) + c_1(z)x + \cdots + c_{n-1}(z)x^{n-1})^{p^i} \pmod{(f(x), \varphi(z))} \\
&\equiv (c_0(z)^{p^i} + c_1(z)^{p^i}x^{p^i} + \cdots + c_{n-1}(z)^{p^i}(x^{n-1})^{p^i} \\
&\equiv (c_0(z^{p^i}) + c_1(z^{p^i})x^{p^i} + \cdots + c_{n-1}(z^{p^i})(x^{p^i})^{n-1} \\
&\equiv c_0(\psi) + c_1(\psi)h_i(x) + \cdots + c_{n-1}(\psi)h_i(x)^{n-1}
\end{aligned}$$

which can be computed with n modular polynomial compositions over $\mathbb{F}_p$ and then one over $\mathbb{F}_q$ (K. and Shoup 1995).