Finish AKS primality Test

Given. n.

We check that for all

$a \leq A$ = polylog(n)

$r \leq R$ = polylog(n)

$$(X+a)^n \equiv X^n + a \pmod{n,\ X^r - 1}$$

WTS: If this passes, then n is a prime power.

Assume Test passes, $p \mid n$ be a prime, $n \neq p^k$.

Want to get a contradiction.

Fix r s.t. $\mathrm{ord}(n) \bmod r > \log^2 n$. $r \nmid p, r \nmid p-1$

Defn $f(x) \in \mathbb{F}_p[X]$ and $m \in \mathbb{N}$

commute if

for all $\omega \in H_r = \{ r^{th}$ roots of 1 in

$\mathbb{F}_p\}$

$$f(\omega^m) = (f(\omega))^m .$$

By the test passing, we know:

$\{X, X+1, X+2, \ldots, X+A\}$ all

commute with n.

Also $f(X)$ and p commute for all $f(X)$.

—

f_1, f_2 commute with $m \Rightarrow$

$f_1 \cdot f_2$ commutes with m

f commutes with $m_1, m_2 \Rightarrow$

f commutes with $m_1 \cdot m_2$.

$\Rightarrow$ Any $f(X) = \prod_{i=1}^{\ell} (X + a_i) \quad \{a_i \le A\}$

and any $n^j p^k$ commute.

$$\left[\forall\, \omega \in H_r. \quad \prod_{i=1}^{\ell} (\omega^{n^j p^k} + a_i) = \left(\prod_{i=1}^{\ell} (\omega + a_i) \right)^{n^j p^k} \right]$$

$$S = \left\{ \prod_{i=1}^{\ell} (X + a_i) \;:\; \text{each } a_i \le A \right\}$$

$$T = \left\{ n^j p^k \;:\; j, k \ge 0 \right\}$$

$$\overline{T} = T \bmod r = \langle n, p \rangle \subseteq \mathbb{Z}_r^*. \quad t = |\overline{T}|.$$

Lemma 1 $\exists \; m^* \le n^{2\sqrt{t}}$ s.t.
$\forall\, \omega \in H_r$, $\forall\, f(X) \in S$,
$f(\omega)^{m^*} = 1$

Proof There are 2 distinct (j, k),

$(j', k') \quad \in [\lceil \sqrt{t} \rceil] \times [\lfloor \sqrt{t} \rfloor]$ s.t.

$$n^{j} \rho^{k} \equiv n^{j'} \rho^{k'} \pmod{p}.$$
$$m \qquad\qquad m'$$

Since $n \neq p^{u}$, $n^{j} \rho^{k} \neq n^{j'} \rho^{k'}$
$m \neq m'$.

Take $f(X) \in S$, $\omega \in H_n$.

$$f(\omega)^{m} = f(\omega^{m})$$
$$= f(\omega^{m'}) = f(\omega)^{m'}.$$

So $f(\omega)^{m-m'} = 1$.

Take $m^{*} = m - m'$.

Lemma 2

$$\left| \{ f(\omega) : \quad \omega \in H_n, f(X) \in S \} \right|$$
$$\gg 2^{t}.$$

Proof

Fix ω_0 a primitive n^{th} root of 1.

If $f(\omega_0) = g(\omega_0)$

for $f(x), g(x) \in S$.

Then for any $m \in \overline{T}$

$$f(\omega_0^m) = \left(f(\omega_0)\right)^m$$

$$= \left(g(\omega_0)\right)^m$$

$$= g(\omega_0^m)$$

$\Rightarrow$ $f(x)$ and $g(x)$ agree on all $\{\omega_0^m : m \in \overline{T}\}$.

i.e. on t distinct pts.

$\Rightarrow$ $f(x) = g(x)$ if both were degree $< t$.

All elements of S of degree $< t$

evaluate to "distinct" values on ω_0.

$$\left| \{ f(\omega) : \omega \in H_n, f(X) \in S \} \right|$$

$$\geq \binom{A+t}{t} \geq 2^t.$$

Since $A \gg t$.

But $2^t > n^{\sqrt{t}}$ (since $t > \log^2 n$, by choice of n)

Contradiction.

---

a,

$$(X+a)^n \equiv X^n + a.$$

$$(\omega + a)^n = \omega^n + a$$

$= \alpha + a$

$(\alpha + a)^n \equiv \alpha^n + a$

$(\omega + a)^{n^2} = \omega^{n^2} + a$

Discrete log

Group G.

Given $g \in G$ and some $x \in G$

s.t. $x = g^i$ for some i,

Find $i := \log_g x$

$G = \mathbb{Z}_{100} = \{0, 1, \ldots, 99\}$ with + mod 100

$g = 1$.

$x = 73$

$\log_g x = 73$.

$G = \mathbb{Z}_{101}^* = \{1, 2, \ldots, 100\}$ with $\times$ mod 101.

$\cong$

$\mathbb{Z}_{100}$.

$g = 7$.

$x = 53$

$\log_g x = ?$ Much more difficult.

How we have access to the group is the whole issue.

$G = \langle g \rangle$, $|G| = n$.

Naive discrete log:

takes n time.
(group operations).

Baby step, Giant steps:

Compute

$$\{ g, g^{\sqrt{n}}, g^{2\sqrt{n}}, \ldots, g^{(\sqrt{n}-1)\cdot\sqrt{n}} \}$$

$$\left\{ x, \frac{x}{g}, \frac{x}{g^2}, \frac{x}{g^3}, \ldots, \frac{x}{g^{\sqrt{n}-1}} \right\}$$

If $x = g^i = g^{a\sqrt{n}+b}$

where $a, b \leq \sqrt{n}$.

then $g^{a\sqrt{n}} = \frac{x}{g^b}$.

So finding a collision gives us the discrete log.

Takes $O(\sqrt{n}\ \text{polylog}(n))$ time

(Sort first list + binary search).

Downside: Requires $\Omega(\sqrt{n})$ space too.

Pollard rho (ρ) algorithm

$O(\sqrt{n}\ \text{polylog}\, n)$ time

$O(\text{polylog } n)$ space.

Discrete log in $\mathbb{Z}_p^*$.

Thm Given g, a generator of $\mathbb{Z}_p^*$ for prime p, we can preprocess and store $B = 2^{O(\sqrt{\log p}\ \text{polyloglog}\, p)}$ bits, and then given any $x \in \mathbb{Z}_p^*$, can compute $\log_g x$ in time $O(B)$.

Based on Smooth numbers.

Defn: A number n is y-smooth if all prime factors of n are $< y$.

Plan: If $x \in \mathbb{Z}_p^*$ as viewed as a natural number is y-smooth, then we can factor

$$x = \prod_{q<y} q^{e_x(q)}$$

and $$\log_g x = \sum_{q<y} e_x(q) \log_g q$$

Claim: If we have an algorithm A such that

$$\Pr_x \left[A(x) = \log_g x \right] \geq \epsilon$$

$x \in \mathbb{Z}_p^*$ $\quad$ g $\quad$ —

Then $\exists$ algorithm A^* s.t.

$\forall x$ $\quad$ $\Pr_{\substack{\text{randomness} \\ \text{of } A^*}} \left[A^*(x) = \log_g x \right] \geq 0.99.$

and $\quad$ runtime $(A^*) = \text{poly}\left(\text{runtime}(A), \frac{1}{\epsilon}\right).$

—

Proof: $A^*(x)$: Repeat $\left(\frac{1}{\epsilon}\right)$ times:
Pick random $j \in [p-1]$
Let $u = A(x \cdot g^j)$
If $g^u = x \cdot g^j$
output $u - j$.