Pollard p−1 factoring

If p=prime $a^{p-1} \equiv 1 \pmod p$

$\mathbb{Z}/n\mathbb{Z} \cong \mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/q\mathbb{Z}$

$n = pq$ Take a mod n.

$a \xrightarrow{\text{square}} a^2 \xrightarrow{\text{cube}} a^{3\cdot 2} \xrightarrow{4^{th}\text{ pow}} a^{4!} \xrightarrow{5^{th}\text{ pow}} a^{5!} \longrightarrow \cdots a^{B!}$

behind scenes

$a \longrightarrow a^2 \longrightarrow a^{3!} \longrightarrow a^4 \longrightarrow a^{5!} \longrightarrow \cdots a^{B!}$ mod p

$a \longrightarrow a^2 \longrightarrow a^{3!} \longrightarrow a^4 \longrightarrow a^{5!} \longrightarrow \cdots a^{B!}$ mod q

So try $\gcd(a^{B!}-1, n)$ "detector" of whether $a^{B!} \equiv 1 \pmod p$

If $p-1$ divides $B!$ but $q-1$ does not divide $B!$

$a^{B!} \equiv 1 \pmod p$ $\quad p \mid a^{B!}-1$

maybe $a^{B!} \not\equiv 1 \pmod q$ $\quad q \nmid a^{B!}-1$

then $p \mid \gcd(a^{B!}-1, n)$ but probably $q \nmid \gcd(a^{B!}-1, n)$.

$\Rightarrow$ get a factor.

Core idea: Do arithmetic mod $n = pq$

Behind scenes, this is arithmetic in $\mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/q\mathbb{Z}$

Try to "catch" a difference in behaviour mod p & mod q.

(in our case $\longrightarrow 1 \bmod p$, $\longrightarrow \neq 1 \bmod q$)

Works well when

$p-1$ is smooth & $q-1$ has a large prime factor

$\Rightarrow$ probability is high that $\begin{cases} p-1 \mid B! \\ q-1 \nmid B! \end{cases}$.

In practice, do a gcd @ each step of chain.

(A) How well this works depends on the factorizations of $p-1$, $q-1$.

$\|$ $\qquad$ $\|$

$|(\mathbb{Z}/p\mathbb{Z})^*|$ $\qquad$ $|(\mathbb{Z}/q\mathbb{Z})^*|$

So for RSA, choose p, q so that $p-1, q-1$ <u>not</u> smooth.

Multiplicative group mod n

calc's → $a \in \mathbb{Z}/p\mathbb{Z}$ (hidden)

$a \in \mathbb{Z}/n\mathbb{Z}$

→ $a \in \mathbb{Z}/q\mathbb{Z}$

$\mathbb{Z}/n\mathbb{Z} \cong \mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/q\mathbb{Z}$

Elliptic Curve E mod n

$E \bmod n$

$P = (x, y)$

→ $E \bmod p$, P_1

→ $E \bmod q$, P_2

visible: $P \xrightarrow{\times 2} 2P \xrightarrow{\times 3} 3!P \xrightarrow{\times 4} 4!P \xrightarrow{\times 5} 5!P \longrightarrow \cdots$

hidden: $(P_1, P_2) \longrightarrow (2P_1, 2P_2) \longrightarrow \cdots$ ($\bmod p$, $\bmod q$)

Detector should go off when $\begin{cases} P_1 = \infty \bmod p \\ P_2 \neq \infty \bmod q \end{cases}$

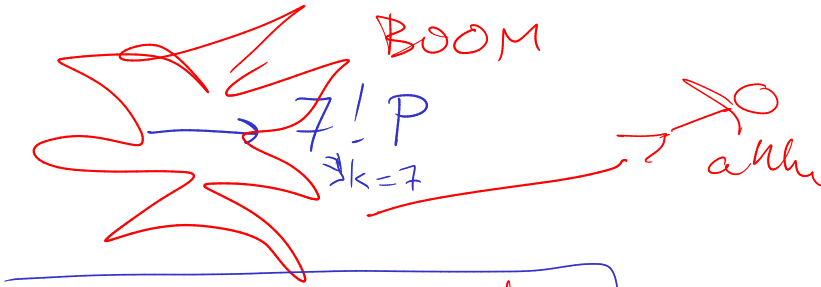
↳ the computation to compute $(k-1)!P$ times k to get $k!P$ will crash / fail.

Example. $n = 18923$

Choose $P = (0,1)$ and $y^2 = x^3 + x + 1$ ← choose curve to contain the point.

$$E: y^2 = x^3 + x + 1 \pmod{n}$$

Compute $P \longrightarrow 2P \longrightarrow 3!P \longrightarrow \cdots$



17272 is not invertible mod n

Compute $\gcd(17272, n) = 127$.

Notes: (1) Scalar mult. on E needs to be fast.

$\Rightarrow$ use successive or double-and-add.

eg. $7P = 4P + 2P + P$

successive doubling: $P \longrightarrow 2P \longrightarrow 4P$

(2) Advantage: order of P mod p divides the # of pts on E mod p, which varies as you choose random curves.

We get lucky when it is smooth.

(3) Runtime: Optimizing B! stopping pt results in subexponential runtime.

(depends on distribution of smooth #s)

$$e^{c\sqrt{(\log p)(\log\log p)}}$$ where p = smallest prime factor of n

(4) Takehome for RSA: keep both primes big.