RSA

en-/decrypt $m^e \bmod pq \to c$

$p, q \sim 5\text{--}$ bits $c^d \bmod pq \to m$

complexity $\sim n^2$ or n^3 operations for n-bit key

factoring seems to need factory & batt $\sim e^{\sqrt[3]{cn(\log n)^2}}$

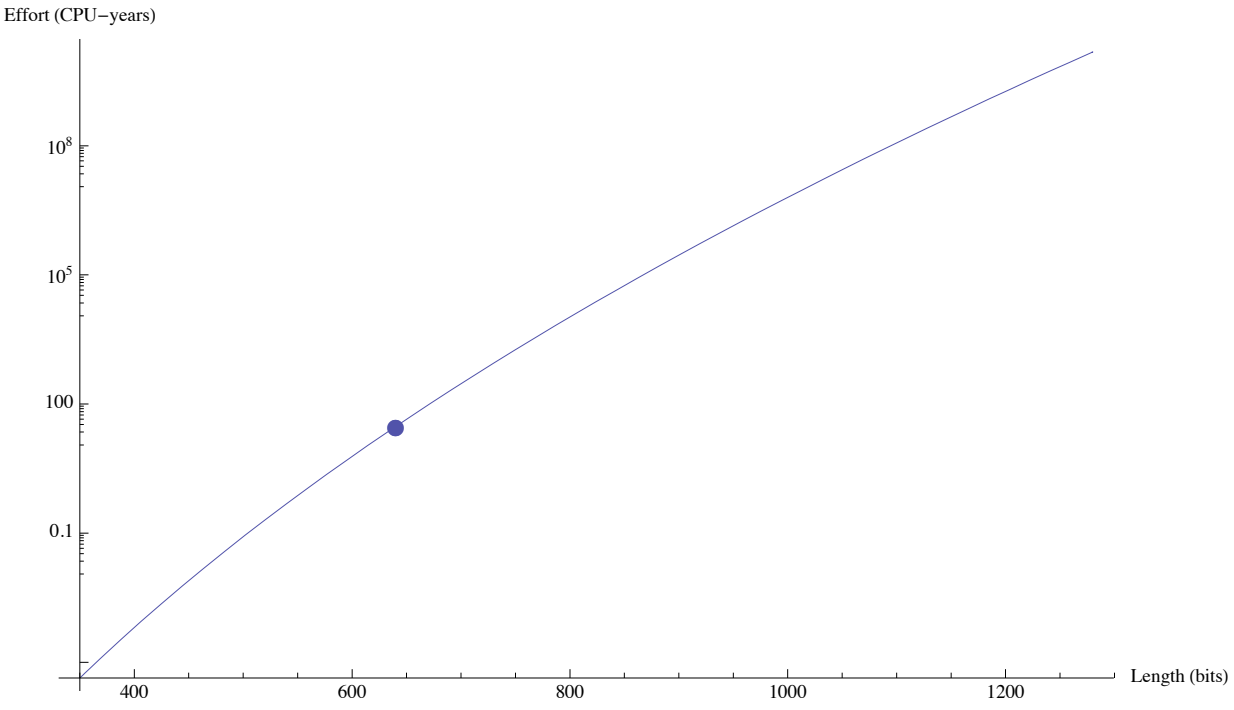
→ how?

"RSA-640", a 640 bit (193-digits) number was factored in 2005 using 30 CPU-years on a cluster of 2.2 GHz Opterons (5 months elapsed). The graph below extrapolates this to other lengths, assuming complexity scales as:

$$e^{\sqrt[3]{\frac{64}{9} n \log^2(n)}}$$

In[150]:=
```
f[n_] := Exp[(64 / 9 * n * Log[n] ^ 2) ^ (1 / 3)]
Show[LogPlot[f[n] / f[640] * 30, {n, 350, 640 * 2},
  AxesLabel → {"Length (bits)", "Effort (CPU-years)"}],
 ListLogPlot[{{640, 30}}, PlotMarkers → {Automatic, Medium}]]
```

Out[151]=



If you're worried about someone spending 30 CPU-years to crack your email chatter, what could you do? *Double your key length.* That will slow en-/decryption by < 8 x (they're sub-cubic algorithms), but slow factoring by 10^8 x.

But a breakthrough in factoring remains possible

Or a breakthrough in quantum computing

$$x^{13} = ((\underbrace{x^2 \cdot x}_{x^3})^2)^2 \cdot x$$

x^3

x^6

x^{12}

x^{13}

$x^{(2^k)}$ takes k squarings

```
pow(x, n, m)
    if n=0 return 1
    y = pow(x, ⌊n/2⌋, m)
    y = y^2 mod m
    if n is odd  y = y·x mod m
    return y
```

mod

Time: linear in length of n

Primality Testing

if n is prime $a^{n-1} \equiv 1 \pmod{n}$

if $a^{n-1} \not\equiv 1 \pmod{n}$ then n is composite

if n is composite then usually $a^{n-1} \not\equiv 1 \bmod n$

for $a = 2$ there are 22 composites $n < 10{,}000$ for which $2^{n-1} \equiv 1 \bmod n$

Try other a?

n is a Carmichael number if it's composite and $a^{n-1} \equiv 1 \pmod{n}$ for all a.

255 c.n.'s $< 10^8$

composite $\equiv$ "has a factor"

But here (and in many other algorithms) simple $\neq$ best!

Other properties might help?

$x^2 = 1 \Rightarrow x = \pm 1$

if p is prime

$(x^2 \equiv 1) \bmod p$ then $x \equiv 1 \pmod p$

or $x \equiv -1 \pmod p$

~~If $p \mid n$~~

These are the only roots.

$x^2 \equiv 1 \bmod p$

$p \mid (x^2 - 1)$

$p \mid (x-1)(x+1)$

either $p \mid (x-1)$ or $p \mid (x+1)$

$x \equiv 1$ or $x \equiv -1 \pmod p$

$4^2 \equiv 1 \pmod{15}$

$4 \not\equiv 1, 4 \not\equiv -1 \pmod{15}$

$\therefore$ 15 is composite.

∀ composites n (even Carmichael)

① ∃ a s.t. either

$$a^{n-1} \not\equiv 1 \pmod{n}$$

or

$$y^2 \equiv 1 \pmod{n} \text{ for}$$

some $y \neq \pm 1 \pmod{n}$

turns up during

pow(a, n-1, n).

② Furthermore $\geq \frac{1}{2}$ of

a's in $1 \leq a \leq n-1$

have this property.

Miller-Rabin Primality Algorithm

```
repeat k times:
    pick random a
    try above
    if "composite" say so & quit
end
Declare n "probably Prime"
```

Miller-Rabin is fast and simple - widely used.

It has the perhaps disturbing feature that numbers it declares to be "prime" may (with some small probability) actually be composite. But for, say, $k = 100$, this error probability is $\leq 2^{-100}$; any algorithm run on a real computer in the real world with real cosmic rays, eg., might also give wrong answers with some small probability.

Final Note: A deterministic algorithm for primality is now known (Agrawal, Kayal, Saxena, 2002). It's somewhat more complex, and slower so less widely used in practice, despite its obvious theoretical advantage. 3x7