Breaking RSA

Bob chose 2 primes

p, q

$N = pq$

$M = (p-1)(q-1)$

Chooses e s.t.

$\gcd(e, M) = 1$

public key $= (N, e)$

set $d = e^{-1} \bmod M$

Alice sends $c \equiv m^e \bmod N$

Bob computes $m = c^d \bmod N$

Eve wants to Factor N

Naive alg: trial division

$O(\sqrt{N}) = O(p)$

Fermat factorization:

$a^2 - b^2 = (a-b)(a+b)$

$O(\sqrt[4]{N}) = O(\sqrt{p})$

Quadratic Sieve

$O\left(e^{\sqrt{\log n \log\log n}}\right)$

General Number Field Sieve

$O\left(e^{\sqrt[3]{64/9}\, \ln(n)^{1/3} \ln(\ln(n))^{2/3}}\right)$

faster if $n \geq 10^{100}$

Pollard's p-1 method

$O(\sqrt[4]{N})$

$N = pq$

suppose large L s.t.

$p-1 \mid L$

$q-1 \nmid L$

$L = i(p-1)$

$L = j(q-1) + k$

$k \neq 0$

If p, q ta, then

$$a^L = a^{i(p-1)} = (a^{p-1})^i \equiv 1^i \equiv 1 \bmod p$$

so $p \mid a^L - 1$

$$a^L = a^{j(q-1)+k} = a^k \cdot a^{j(q-1)} \equiv a^k \bmod q$$

probably $a^k \not\equiv 1 \bmod q$

$q \nmid a^L - 1$

Then

$\gcd(a^L - 1, N) = p.$

Euclidean Algorithm:

$a > b > 0$ want $\gcd(a, b)$

1) set $r_0 = a$, $r_1 = b$

2) for $i \geq 1$, divide r_{i-1} by r and set r_{i+1} equal to remainder

3) repeat until $r_{k+1} = 0$.

then $\gcd(a, b) = r_k$

Sketch of pf:

$$\gcd(a, b) = \gcd(b, a-b)$$
$$= \gcd(b, r_2)$$
$$= \gcd(r_2, r_3)$$
$$= \gcd(r_3, r_4) \sim$$

Ex: $\gcd(20, 78)$

$r_0 = 78$, $r_1 = 20$

$78 = 3 \cdot 20 + 18$ $\quad r_2 = 18$

$20 = 1 \cdot 18 + 2$ $\quad r_3 = 2$

$18 = 9 \cdot 2 + 0$ $\quad r_4 = 0$

So $\gcd(20, 78) = r_3 = 2$.

$\gcd(94012, 33396) = 4$

$94012 = 2 \cdot 33396 + 27220$

$33396 = 1 \cdot 27220 + 6176$

$r_4 = 2576$

$r_5 = 1144$

$r_6 = 288$

$r_7 = 4$

$r_8 = 0$

Fact (Lamé):

EA $\leq \log_2(ab)$ steps

$\leq 5 \log_{10}(b)$ steps

So we want L s.t.

$p-1 \mid L$

$q-1 \nmid L$.

Let's hope we get lucky

maybe $p-1$ only has small prime factors

then $p-1 \mid n!$

for small n.

So. compute $\gcd(a^{n!}-1, N)$

for small N.

$10! = 3,628,800$

$2^{10!}$ has a million digits

3.6 m bit number

$2^{100!}$ has 10^{157} digits

Cannot write down $2^{100!}$

But we can work mod N.

And: at each step,

$2^{n!} = \left(2^{(n-1)!}\right)^n$.

Algorithm:

set $a_1 = 2$.

for $i > 1$

- Let $a_i = a_{i-1}^i \bmod N$ $= 2^{i!}$
- compute $d = gcd(a_i - 1, N) = gcd(2^{i!} - 1, N)$
- if $1 < d < N$, then d is a prime factor of N
- else repeat.

computing $a^{n!} \bmod N$ is $O(2n \log_2(N))$

So how big is n?

need $p-1 \mid n!$

happens when $n \geq$ largest prime factor of $p-1$.

hoping $p-1$ factors into small primes

so n is small

worst case: $p-1 = 2 \cdot$ prime

then $n = \frac{p-1}{2}$

$O(p \log(N)) = O(\sqrt{N} \log(N))$

But: Bob can check if $p-1$ factors into small primes!

So Bob can choose p, q to not split into small primes

Ex: factor 1411

$2^{2!} = 2^2 = 4$

$2^{2!} - 1 = 3$

$\gcd(3, 1411) = 1$

$2^{3!} = 2^6 = 64$

$\gcd(63, 1411) = 1$

$2^{4!} = 2^{24} \equiv 426 \bmod 1411$

$\gcd(425, 1411) = 17$

$2^{5!} = (426)^5 \equiv 1276 \bmod 1411$

$\gcd(1276, 1411) = 1.$

$2^{6!} \equiv 1276^6 \equiv 783 \bmod 1411$

$\gcd(783, 1411) = 1$

$2^{8!} \equiv 783^{56} \equiv 732$

So $17 \mid 1411$ $\quad$ $1411 = 17 \cdot 83.$

Discrete log

Factoring with DL:

want to factor N.

1) choose random $a < N$

2) compute $\gcd(a, N) = d$

3) if $d > 1$, done, $b \leftarrow d/N$

4) else, find $r = \log_a 1 \bmod N$

5) if r is odd, start over

6) if $a^{r/2} \equiv \pm 1 \bmod N$, start over

7) else $a^{r/2}+1$, $a^{r/2}-1 \not\equiv \pm 1 \bmod N$.

$(a^{r/2}+1)(a^{r/2}-1) = a^{\wedge}-1 \equiv 0 \bmod N$.

8) compute $\gcd(a^{r/2}-1, N)$

$\gcd(a^{r/2}+1, N)$.

both factors of N.