RSA (Public-key cryptosystem)

- Cryptology is a major consumer of number theory.

- Primality & integer factoring appear in a cryptosystem by Rivest, Shamir & Adleman (1977).

Preprocessing:

1) Carefully choose prime $p \neq q$.
2) $n := p \cdot q$ & $\phi(n) := (p-1) \cdot (q-1)$.
3) Choose $1 < e < \phi(n)$ coprime to $\phi(n)$ & n.

   (n, e) is the public key.

4) $d := e^{-1} \bmod \phi(n)$ is the private key.

Encryption: $m \mapsto m^e \bmod n$.

Decryption: $c \mapsto c^d \bmod n$.

▷ $m \mapsto m^e \mapsto (m^e)^d \equiv m^{1+k\cdot\phi(n)} \equiv m \pmod{n}$.

OPEN: Given (n,e), is there an efficient way to compute $e^{-1} \bmod \phi(n)$ (or $c^{1/e} \bmod n$)?

Exercise: $\phi(n)$ & factoring n are equivalent up to randomized poly-time.

▷ Integer factoring cracks RSA.

(RSA problem) ▷ e-th root finding (i.e. $c^{1/e} \bmod n$) also cracks RSA.

OPEN: Is RSA problem equivalent to integer factoring up to randomized poly-time?

Integer factoring

- The general algorithms to factor n are slow! Currently, only integers in ≈ 700 bits or ≈ 200 digits could be factored; that too using specialized hardware.

- The best provable complexity known is: Expected time $\exp(O(\sqrt{\lg n \cdot \lg\lg n}))$.

- Heuristically, it is $\exp(O(\lg^{1/3} n \cdot \lg^{2/3} \lg n))$.

- We will use the notation $L_x(\alpha, c) :=$
$$\exp(c \cdot \log^{\alpha} x \cdot \log^{1-\alpha} \log x).$$

[Pomerance, 1989]: The general number field sieve (GNFS) has conjectured complexity
$$L_n(1/3, 2).$$

- Why this 'strange' function $L_x(\alpha, c)$?

Smooth numbers

Defn: • A number n is *y-smooth* if all the prime factors of m are $\leq y$.
• Their *density* is denoted by $\psi(x,y)$ $:= \#\{1 < m \leq x \mid n \text{ is } y\text{-smooth}\}$.

– Asymptotic estimate for $\psi(x,y)$ determines the complexity of advanced factoring algorithms

Thm. (Dickman-de Bruijn '51) $\psi(x,y) \approx x/u^u$, $u := \log_y x$.

Pf idea:

• Consider the regime $u \lesssim t := y/\log y$.

• There are roughly t primes $2 = p_1 < p_2 < \dots < p_t$ below y.

• Any good m can be expressed as $\prod_{i=1}^{t} p_i^{\alpha_i}$.

• Clearly, $\psi(x,y) \geq \#\{\bar{\alpha} \mid \sum_{i=1}^{t} \alpha_i \leq \log_y x\}$

$$\approx \binom{u+t}{u} \approx \left(\frac{t}{u}\right)^u \approx \frac{x}{u^u} \qquad \square$$

- This bound is sensible only when $u^u \ll x$. Thus, y should not be too small.

▷ A useful, tolerable y is $L_x(\alpha, c)$, for constants α, c, with $u^u \approx L_x\left(1-\alpha, \frac{1-\alpha}{c}\right)$.

Pf idea:

- $\log y \approx \log^{\alpha} x \cdot \log^{1-\alpha} \log x$

$$\Rightarrow u = \frac{\log x}{\log y} \approx \frac{1}{c} \cdot \log^{1-\alpha} x \cdot \log^{\alpha-1} \log x$$

$$\Rightarrow u \cdot \log u \approx \frac{1}{c}\left(\log^{1-\alpha} x \cdot \log^{\alpha-1} \log x\right) \cdot (1-\alpha) \cdot \log \log x$$

$$\approx \frac{1-\alpha}{c} \cdot \log^{1-\alpha} x \cdot \log^{\alpha} \log x.$$

$$\Rightarrow u^u \approx L_x\left(1-\alpha, \frac{1-\alpha}{c}\right). \qquad \square$$

▷ Thus, for $y = L_x(\alpha, c)$ the probability of choosing a y-smooth $m \leq x$ is given by

$$\frac{\Psi(x,y)}{x} \approx L_x\left(1-\alpha, -\frac{1-\alpha}{c}\right).$$