Complexity of Random Squares: Test random numbers of size $\mathcal{O}(N)$ how many numbers need to check? Write $L_N(1/2,1) = L_N$ for simplicity. Then probability of random number to be B-smooth, for $B = L_N^c$ is $L_N^{-1/2c}$ (Corollary of Canfield, Erdos, Pomerance). We saw from Exercise 6 that minimum number of relations is $\pi(B)$. From Prime Number theorem we know this is $\approx B/\log B$. In total we want to miminize c in the expression:

$$\frac{L_N^c}{c\log L_N} \cdot \frac{1}{L_N^{-1/2c}} = L_N^{c+1/2c} \cdot \frac{1}{c\log L_N} \tag{1}$$

which implies $c = 1/\sqrt{2}$ and a value of $\sqrt{2}$. So for $B = L_N^{1/\sqrt{2}}$ we need to check approximately $L_N^{\sqrt{2}}$ numbers.
This was improved by quadratic sieve to L_N numbers when size is $\mathcal{O}(\sqrt{N})$ by noting that $L_N^{1/\sqrt{2}} \approx L_{\sqrt{N}}$.

Number Field Sieve: Key observation: further reduce size of numbers to be tested. From $\mathcal{O}(\sqrt{N})$ to $\mathcal{O}(L_N(2/3,c))$ then numbers are B-smooth with high prob. when $B = \mathcal{O}(L_N(1/3,c))$. Hard to choose such numbers directly so work with two factor bases. Choose $m \in \mathbb{N}$ and a monic irreducible polynomial $f(x)$ such that $f(m) \equiv 0 \pmod N$. Then $\mathcal{B}_1$ = primes up to B and $\mathcal{B}_2$ = prime ideals in $\mathbb{Z}[x]/(f(x)) = \mathbb{Z}[\theta]$ for generic root θ (write set).

Definition 1 (Prime ideal) An ideal $\mathcal{I}$ such that $ab \in \mathcal{I} \Rightarrow a \in \mathcal{I}$ or $b \in \mathcal{I}$.

Corresponds to prime numbers property $p|ab \Rightarrow p|a$ or $p|b$.

Look (by sieving) for pairs of integers (a,b) such that simultaneously $a - bm$ factors over $\mathcal{B}_1$ and $(a-b\theta)$ factors over $\mathcal{B}_2$. Then $a-bm = \prod p_i^{e_i}$ and $(a-b\theta) = \prod \mathcal{I}_j^{\ell_j}$; we store exponents as in random squares. When we have enough pairs, we obtain $\prod_{(a,b)\in S}(a-bm) = u^2$ and $\prod_{(a,b)\in S}(a-b\theta) = v^2$ and we can relate those via the homomorphism $\phi : \mathbb{Z}[x]/(f(x)) \to \mathbb{Z}_N$ defined as $\phi(\theta) = m \pmod N$. So $u^2 \equiv \phi(v)^2 \pmod N$ and factor computing $(u-\phi(v), N)$. *Heuristic complexity $L_N(1/3, c+o(1))$ for small $c = (64/9)^{1/3} \approx 1.923$.*

Remark 1 Simplified versions due to several issues with e.g. principal ideals (single element) and units.

RSA Variants

1. Speedup Encryption: Choose small e makes encryption faster. First idea: 3 (ask). Possible insecurity. Best choice $e = 65537 = 2^{16} + 1$ fast (4 squarings).

Hastad (broadcast) attack: Suppose $e = 3$ and we broadcast same message to three users with public keys N_1, N_2, N_3. We require $m < N_i$ for all i. Then for $i = 1,2,3$ we have $c_i \equiv m^e \pmod{N_i} = m^3 \pmod{N_i}$. Use CRT and find $c \equiv m^3 \pmod{N_i}$ with $c < \tilde{N} = N_1N_2N_3$. It follows that $c = m^3$ over $\mathbb{Z}$ and we can take cubic root.
Can be prevented using padding scheme.

Franklin-Reiter (Related Message) attack: Suppose $e = 3$ and same public key N. Suppose $c_1 = Enc_N(m)$ and $c_2 = Enc_N(m+a)$. Then m is common root of polynomials $f_1(x) = x^e - c_1$

and $f_2(x) = (x+a)^e - c_2$. This means $(x-m)$ divides both. So we can run Euclidean algorithm on $f_1(x), f_2(x)$ in $\mathbb{Z}_N[x]$ and get common factor $(x-m)$.
Hint: can be extended.

2. Speedup Decryption: Choose d small first, and then calculate $e = d^{-1} \pmod{\varphi(N)}$.

Wiener attack: Successful for $d < 1/3N^{1/4}$. Suppose pk is $N = pq$ with $p < q < 2p$. Then $p < \sqrt{N}$. Now, $ed \equiv 1 \pmod{\varphi(N)}$ so $ed = 1 + k\varphi(N)$ for some integer k, so we can write:

$$\left|\frac{k}{d} - \frac{e}{N}\right| = \left|\frac{Nk - ed}{Nd}\right| = \left|\frac{Nk - (1 + k\varphi(N))}{Nd}\right| = \left|\frac{k(N - \varphi(N)) - 1}{Nd}\right| < \frac{3k\sqrt{N}}{Nd} < \frac{3k}{d\sqrt{N}} \qquad (2)$$

since $N - \varphi(N) < 3\sqrt{N}$ (explain). Now, $k < d$ implies $3k < 3d < N^{1/4}$ so in total $< 1/3d^2$. Hence the public information e/N is a good approximation of k/d. Apply theorem of Hardy and Wright about continued fractions. We can then find k/d as a convergent and guess the value of $\varphi(N) = ed - 1/k$, then verify $(x-p)(x-q) = x^2 - (p+q)x + pq = x^2 - (N - \varphi(N) + 1)x + N$ solving the equation gives the factorization.

2a. CRT decryption exponents: Instead of using d we can calculate $d_p \equiv e^{-1} \pmod{p-1}$ and $d_q \equiv e^{-1} \pmod{q-1}$ and compute $m_p \equiv c_p^{d_p} \pmod{p}$ and $m_q \equiv c_q^{d_q} \pmod{q}$. Use CRT for finding solution modulo N. This way we lower the cost from $\kappa \log(N) M(\log(N))$ (cost of exp modulo N to a power of size $\approx N$) to $2\kappa \log(N)/2 M(\log(N)/2)$, that is almost a factor of 3 in practice.
We assume cost of CRT is negligible.

2b. Multiprime RSA (Collins, Hopkins, Langford, Sabin): Build on previous idea and suppose $N = p_1 \cdot \cdots \cdot p_k$ each prime of size γ/k. Cost of CRT is approx k exponentiations to powers γ/k modulo γ/k so $\approx 1/k^{1.58}$. Ex speedups $3 \to 5.7 \to 8.9$.
Limit as how k can be because of attacks of the first class.

2c. Takagi RSA: Again build on previous idea and suppose $N = p^r q$ for $r > 1$. Compute $d_p \equiv d \pmod{p-1}$ and $d_q \equiv d \pmod{q-1}$ and compute $m_p \equiv c_p^{d_p} \pmod{p}$ and $m_q \equiv c_q^{d_q} \pmod{q}$. Then use Hensel lifting: if we have $m_i^e \equiv c \pmod{p^i}$ we write $m_{i+1} = m_i + xp^i$ and then $m_{i+1}^e = (m_i + xp^i)^e \equiv m_i^e + x(em_i^{e-1})p^i \pmod{p^{i+1}} \equiv c \pmod{p^{i+1}}$. Cost dominated by two exponentiations for m_p and m_q so speedup $2/(r+1)^{2.58}$ so for $r = 2$ about 9 times faster than naive RSA (1.6 times faster than 3-prime RSA) and for $r = 3$ about 18 times (2 times faster than 4-prime RSA).
Only efficient for small e, otherwise Hensel lifting stage not negligible anymore.