CSE 521: Design and Analysis of Algorithms I

Randomized Algorithms: Primality Testing

Paul Beame

Randomized Algorithms

- **QuickSelect** and **Quicksort**
  - Algorithms' random choices make them fast and simple but don't affect correctness
  - Not only flavor of algorithmic use of randomness
- **Def:** A randomized algorithm **A** computes a function **f** with error at most ε iff
  - For **every** input **x** the probability over the random choices of **A** that **A** outputs **f(x)** on input **x** is $\geq$ **1-** ε
- Error at most 2^{-100} is practically just as good as **0**
  - Chance of fault in hardware is larger

Primality Testing

- Given an n-bit integer N determine whether or not N is prime.
- Obvious algorithm: Try to factor N
  - Try all divisors up to $N^{1/2} \leq 2^{n/2}$.
  - Best factoring algorithms run in $\geq 2^{n^{1/3}}$ time
- Rabin-Miller randomized algorithm
  - If N is prime always outputs “prime”
  - If N is composite
    - outputs “composite” with probability $1\text{-}2^{-2t}$
    - outputs “prime” with probability 2^{-2t}
- [AKS 2002] Polynomial-time deterministic algorithm.
  - Much less efficient, though.

Rabin-Miller Algorithm

- If N is even then output "prime" if $N=2$ and "composite" otherwise and then halt
- Compute k and d such that $N-1=2^k d$ where d is odd
- For $j=1$ to t do
  - Choose random a from $\{1,\ldots,N-1\}$
  - Compute $b_0=a^d \bmod N$ using powering by repeated squaring
  - For $i=1$ to k do
    - Compute $b_i=b_{i-1}^2 \bmod N = a^{2^i d} \bmod N$
    - If $b_i=1$ and $b_{i-1} \neq \pm 1$ output "composite" and halt
  - If $b_k = a^{N-1} \bmod N \neq 1$ output "composite" and halt
- Output "prime"

- Running time: $O(tn)$ multiplications $\bmod N$

Rabin-Miller analysis

- We will prove slightly weaker bound:
  - If N is prime always outputs "prime"
  - If N is composite
    - outputs "composite" with probability $1-2^{-t}$
    - outputs "prime" with probability 2^{-t}
- Whenever output is "composite" N is composite:
  - Fermat's Little Theorem: If N is prime and a is in $\{1,\dots,N-1\}$ then $a^{N-1} \bmod N = 1$
    - So $a^{N-1} \bmod N \neq 1$ implies N is composite
  - If $b_i = b_{i-1}^2 \bmod N = 1$ then N divides $(b_{i-1}^2 - 1) = (b_{i-1} - 1)(b_{i-1} + 1)$ so if N is prime then N divides $(b_{i-1} - 1)$ or $(b_{i-1} + 1)$ and thus $b_{i-1} = b_{i-1} \bmod N = \pm 1$
    - So $b_i = 1$ and $b_{i-1} \neq \pm 1$ implies N is composite

Some observations

- Let m be any integer > 0
- If $\gcd(a,N)>1$ for $0<a<N$ then N is composite but also $\gcd(a^m,N)>1$ so $a^m \bmod N \neq 1$
  - Algorithm will test $m=N-1$ and output "composite"
- Write $Z_N^*=\{a \mid 0<a<N \text{ and } \gcd(a,N)=1\}$
  - Euclid's algorithm shows that every b in Z_N^* has an inverse b^{-1} in Z_N^* such that $b^{-1} b \bmod N = 1$
- Let $G_m=\{a \text{ in } Z_N^* \mid a^m \bmod N = 1\}$
- Claim: If there is a b in Z_N^* but not in G_m then

$$|G_m| \leq |Z_N^*|/2.$$

Some observations

- $Z_N^* = \{a \mid 0<a<N$ and $\gcd(a,N)=1\}$
- Let $G_m = \{a$ in $Z_N^* \mid a^m \bmod N = 1\}$
- Claim: If there is a b in Z_N^* but not in G_m then

  $|G_m| \le |Z_N^*|/2$.

  - Consider $H_m = \{ba \bmod N \mid a$ in $G_m\} \subseteq Z_N^*$.
  - Then $|H_m|=|G_m|$ since $ba_1=ba_2 \bmod N$ implies $a_1=a_2 \bmod N$
  - Also for c in H_m, $c=ba \bmod N$ for some a in G_m.

    so $c^m \bmod N=(ba)^m \bmod N$

    $= b^m a^m \bmod N = b^m \bmod N \neq 1$.

Carmichael Numbers

- So… if there is even one **a** such that a^{N-1} mod $N \neq 1$ then **N** is composite and at least half the possible **a** also satisfy this and the algorithm will output "composite" with probability $\geq \frac{1}{2}$ on each time through the loop
  - Chance of failure over **t** iterations $\leq 2^{-t}$.
- Odd composite numbers (e.g. **N=361**) that have a^{N-1} mod $N=1$ for all **a** in Z_N^* are called Carmichael numbers
- Fact: Carmichael numbers are not powers of primes
  - Only need to consider the case of $N=q_1q_2$ where $\gcd(q_1,q_2)=1$

Rabin-Miller analysis

- Need the other part of the Rabin-Miller test
  - If $b_i = a^{2^i d}$ mod $N = 1$ and $b_{i-1} = a^{2^{i-1} d}$ mod $N \neq \pm 1$ output "composite"
  - Chinese Remainder Theorem:
    - If $N = q_1 q_2$ where $\gcd(q_1, q_2) = 1$ then for every r_1, r_2 with $0 \le r_i \le q_i - 1$ there is a unique integer M in $\{0, \ldots, N-1\}$ such that M mod $q_i = r_i$ for $i = 1, 2$.

      (One-to-one correspondence between integers M and pairs r_1, r_2)
  - $M = 1 \leftrightarrow (1,1)$, $M = -1 = N-1 \leftrightarrow (q_1 - 1, q_2 - 1) = (-1, -1)$
  - Other values of M such that M^2 mod $N = 1$ correspond to pairs $(1, -1)$ and $(-1, 1)$

Finishing up

- Consider the largest i such that there is some a_1 in Z_N^* with $a_1^{2^{i-1}d}$ mod $N=-1$ and let $r_i= a_1$ mod q_i
- Since $a_1 \neq 1$, $(r_1,r_2) \neq (1,1)$. Assume wlog $r_1 \neq 1$.
- Let $G=\{a$ in $Z_N^* \mid a^{2^{i-1}d}$ mod $N=\pm 1\}$
- By Chinese Remainder Theorem consider b in Z_N^* corresponding to the pair $(r_1,1)$.
  - Then $b^{2^i d}$ mod $q_1=1$ and $b^{2^i d}$ mod $q_2=1$ so $b^{2^i d}$ mod $N=1$
  - But $b^{2^{i-1}d}$ mod $q_1=-1$ and $b^{2^{i-1}d}$ mod $q_2=1$ so $b^{2^{i-1}d}$ mod $N \neq \pm 1$
- By similar reasoning as before every element of $H=\{ba \mid a$ in $G\}$ is in Z_N^* but not in G so $|G| \leq |Z_N^*|/2$ and the algorithm will choose an element not in G with probability $\geq$ ½ per iteration and output "composite" with probability $\geq 1-2^{-t}$ overall

Relationship to Factoring

- In the second case the algorithm finds an x such that $x^2 \bmod N = 1$ but $x \bmod N \neq \pm 1$
  - Then N divides $(x^2-1)=(x+1)(x-1)$ but N does not divide $(x+1)$ or $(x-1)$
  - Therefore N has a non-trivial common factor with both $x+1$ and $x-1$
  - Can partially factor N by computing $\gcd(x-1,N)$
- Finding pairs x and y such that $x^2 \bmod N = y^2$ but $x \neq \pm y$ is the key to most practical algorithms for factoring (e.g. Quadratic Sieve)

Basic RSA Application

- Choose two random **n**-bit primes **p**, **q**
  - Repeatedly choose **n**-bit odd numbers and check whether they are prime
  - The probability that an **n**-bit number is prime is $\Omega(\mathbf{1/n})$ by the Prime Number Theorem so only $O(\mathbf{n})$ trials required on average
- Public Key is $\mathbf{N}=\mathbf{pq}$ and random **e** in $\mathbf{Z_N}^*$
  - Encoding message **m** is $\mathbf{m}^{\mathbf{e}} \bmod \mathbf{N}$
- Secret Key is $(\mathbf{p},\mathbf{q})$ which allows one to compute $\varphi(\mathbf{N}) = \mathbf{N}-\mathbf{p}-\mathbf{q}+\mathbf{1}$ and $\mathbf{d}=\mathbf{e}^{-1} \bmod \varphi(\mathbf{N})$
  - Decryption of ciphertext **c** is $\mathbf{c}^{\mathbf{d}} \bmod \mathbf{N}$

- Note: Some implementations (e.g. PGP) don’t do full Rabin-Miller test