MATH 314 Spring 2018 - Class Notes

04/16/2018

Scribe: Chance Brilz

Summary: Today we covered four different primality tests, in addition to revisiting Euler's Primality Test.

Main Question: How do we get prime numbers for RSA?

- Ideally we ant prime numbers with around 120 digits
- We can use random number generators to pick random 120 digit numbers.
  - The random number generator must be cryptographically secure.

Steps to pick random prime numbers

1. Ideally we ant prime numbers with around 120 digits
2. We can use random number generators to pick random 120 digit numbers.

Question: How do we know there are enough 120 random prime numbers?

Define: Prime number counting function

$\pi(x) = \#$ of prime numbers less than or equal to x

e.g. $\pi(10) = 4$,$\pi(11) = 5$, $\pi(12) = 5$ ...

Note: # of prime numbers with 120 digits is around $\pi(10^120)$ - $\pi(10^119)$

Prime Number Theorem: $\pi(x) = \frac{x}{\ln(x)}$

Checking Primality

Fermat's Primality Test

Steps:

1. Pick a random a
2. If $a^{m-1} \not\equiv 1$ (mod m) return "Composite"
3. Repeat i times, if the loop finishes, return "Prime"

Solovay-Strassen Primality Test

Steps:

1. Pick a random a in $2 \leq a - 1 \leq n - 1$
2. If $(\frac{a}{n}) \not\equiv a^{\frac{n-1}{2}}$ (mod n), return "Composite"
3. Repeat i times, if the loop finishes, return "Prime"

Miller-Rabin Primality Test

Steps:

1. Write $n - 1 = m * 2^k$, where m is odd
   e.g. $n = 21 => 20 = 5 * 2^2$
2. Pick a randomly, $2 \leq a \leq n - 1$
3. If $b_0 = \pm 1$ (mod n), then n is probably prime
4. For i in 1 up to $k - 1$
   Compute $b_i \equiv b_{i-1}^2$ (mod n)
   If $b_i \equiv 1$ (mod n) return "Composite"
   If $b_i \equiv -1$ (mod n) return "Prime"
5. If the loop finishes, return "Composite"