

MA/CSSE 473

Day 11

Primality Testing

Data Encryption



MA/CSSE 473 Day 11

- I added an article on the Miller-Rabin test to the reading for Day 10.
- HW 5 is due tomorrow.
- HW 6 includes an implementation problem (described in the HW5 document)
 - You can work with another student
 - Start soon
- Exam 1: Tuesday, September 30
 - You may bring your textbook, plus a one-sided 8.5x11 inch piece of paper containing anything that you can read unaided or with normal eyeglasses
- **Student Questions**
- Primality Testing – Miller-Rabin (helps with Carmichael numbers)
- Generating Random Primes
- Cryptography Introduction



Recap: Fermat's Little Theorem

- **Formulation 1:** If p is prime, then for every number a with $1 \leq a < p$, $a^{p-1} \equiv 1 \pmod{p}$
- What does it tell us if p is not prime?



The algorithm (modified)

- To test N for primality
 - Pick positive integers $a_1, a_2, \dots, a_k < N$ at random
 - For each a_i , check for $a_i^{N-1} \equiv 1 \pmod{N}$
 - Use the Miller-Rabin approach, (next slide) so that Carmichael numbers are unlikely to thwart us.
 - If $a_i^{N-1} \pmod{N} \neq 1$, or the Miller-Rabin test produces a non-trivial square root of 1 (mod N)
 - return false
 - return true



Miller-Rabin test

- A Carmichael N number is a composite number that passes the Fermat test for all a with $0 < a < N$
- A way around this (Rabin and Miller):
Note that for some t and u (u is odd), $N-1 = 2^t u$.
- As before, compute $a^{N-1} \pmod{N}$, but do it this way:
 - Calculate $a^u \pmod{N}$, then repeatedly square, to get the sequence
 $a^u \pmod{N}, a^{2u} \pmod{N}, \dots, a^{2^t u} \pmod{N} \equiv a^{N-1} \pmod{N}$
- Suppose that at some point, $a^{2^i u} \equiv 1 \pmod{N}$, but $a^{2^{i-1} u}$ is not congruent to 1 or $N-1 \pmod{N}$
 - then we have found a nontrivial square root of 1 \pmod{N} .
 - We will show that if 1 has a nontrivial square root \pmod{N} , then N cannot be prime



Lemma: Modular Square Roots of 1

- If s is neither 1 or $-1 \pmod{N}$, but $s^2 \equiv 1 \pmod{N}$, then N is not prime
- **Proof** (by contrapositive):
 - $s^2 - 1 \equiv 0 \pmod{N}$
 - $(s - 1)(s + 1) \equiv 0 \pmod{N}$
 - So N divides $(s - 1)(s + 1)$
 - If N is prime, N divides $(s - 1)$ or N divides $(s + 1)$
 - But this contradicts our assumption about s
- This validates the Miller-Rabin test



Example (first Carmichael number)

- $N = 561$. We might randomly select $a = 101$.
 - Then $560 = 2^4 \cdot 35$, so $u=35$, $t=4$
 - $a^u \equiv 101^{35} \equiv 560 \pmod{561}$ (which is -1) (we can stop here)
 - $a^{2u} \equiv 101^{70} \equiv 1 \pmod{561}$
 - ...
 - $a^{16u} \equiv 101^{560} \equiv 1 \pmod{561}$
 - So 101 is not a witness that 561 is composite (we say that 101 is a *liar for 561*, if indeed 561 is composite)
- Try $a = 83$
 - $a^u \equiv 83^{35} \equiv 230 \pmod{561}$
 - $a^{2u} \equiv 83^{70} \equiv 166 \pmod{561}$
 - $a^{4u} \equiv 83^{140} \equiv 67 \pmod{561}$
 - $a^{8u} \equiv 83^{280} \equiv 1 \pmod{561}$
 - So 83 is a witness that 561 is composite.



Accuracy of the Primality Test

- Rabin showed that if N is composite, this test will demonstrate its non-primality for at least $\frac{3}{4}$ of the numbers a that are relatively prime to N , even if a is a Carmichael number.
- Note that $\frac{3}{4}$ is the worst case; randomly-chosen composite numbers have a much higher percentage of witnesses to their non-primeness.
- If we test several values of a , we have a very low chance of flagging a composite number as prime.



Efficiency of the Test

- Testing an n -bit number is $\Theta(n^3)$
- If we use the fastest-known integer multiplication techniques (based on Fast Fourier Transforms), this can be pushed to $\Theta(n^2 \log n \log \log n)$



Testing "small" numbers

- **Wikipedia article on the Miller-Rabin primality test:**
- When the number N we want to test is small, smaller sets of potential witnesses are known to suffice. For example, Jaeschke has verified that
 - if $N < 9,080,191$, it is enough to test $a = 31$ and 73
 - if $N < 4,759,123,141$, it is enough to test $a = 2, 7$, and 61
 - if $N < 2,152,302,898,747$, it is enough to test $a = 2, 3, 5, 7, 11$
 - if $N < 3,474,749,660,383$, it is enough to test $a = 2, 3, 5, 7, 11, 13$
 - if $N < 341,550,071,728,321$, it is enough to test $a = 2, 3, 5, 7, 11, 13, 17$.



Generating Random Primes

- For cryptography, we want to be able to quickly generate random prime numbers with a large number of bits
- Are prime numbers abundant among all integers? Fortunately, yes
- Lagrange's prime number theorem
 - Let $\pi(N)$ be the number of primes that are $\leq N$, then $\pi(N) \approx N / \ln N$.
 - Thus the probability that an n -bit number is prime is approximately $(2^n / \ln(2^n)) / 2^n \approx 1.44 / n$



Random Prime Algorithm

- Pick a random n -bit number N
- Run a primality test on N
- If it passes, output N
- Else repeat the process
- Expected number of iterations is $\Theta(n)$



Cryptography

- I want to transmit a message m to you
 - in a form $e(m)$ that you can readily decode by running $d(e(m))$,
 - but an eavesdropper has little chance of decoding
- Private-key protocols
 - You and I meet beforehand and agree on e and d .
- Public-key protocols
 - You publish an e for which you know the d , but it is very difficult for someone else to guess the d .



RSA Public-key Cryptography

- Rivest-Shamir-Adleman (1977)
 - A reference that you probably have: Mark Weiss, Data Structures and Problem Solving Using Java 7.4
- Consider a message to be a number modulo N , an n -bit number (longer messages can be broken up into n -bit pieces)
- The encryption function will be a bijection on $\{0, 1, \dots, N-1\}$ and the decryption function will be its inverse
- How to pick the N and the bijection?



$$N = p q$$

- Pick any two large primes, p and q , and let $N = pq$.
- **Property:** If e is any number that is relatively prime to $(p-1)(q-1)$, then
 - the mapping $x \rightarrow x^e \pmod{N}$ is a bijection on $\{0, 1, \dots, N-1\}$
 - If d is the inverse of $e \pmod{(p-1)(q-1)}$, then for all x in $\{0, 1, \dots, N-1\}$, $(x^e)^d \equiv x \pmod{N}$.
- We'll apply the property, then prove it.



Public and Private Keys

- The first (bijection) property tells us that $x \rightarrow x^e \pmod N$ is a reasonable way to encode messages, since no information is lost
 - If you publish (N, e) as your *public key*, anyone can encrypt and send messages to you
- The second tells how to decrypt a message
 - Keep your *private key*, d , so that when you receive a message m' , you can decode it by calculating $(m')^d \pmod N$.



Example (from Wikipedia)

- $p=61, q=53$. Compute $N = pq = 3233$
- $(p-1)(q-1) = 60 \cdot 52 = 3120$
- Choose $e=17$ (relatively prime to 3120)
- Compute multiplicative inverse of 17 (mod 3120)
 - $d = 2753$ (evidence: $17 \cdot 2753 = 46801 = 1 + 15 \cdot 3120$)
- To encrypt $m=123$, take $123^{17} \pmod{3233} = 855$
- To decrypt 855, take $855^{2753} \pmod{3233} = 123$
- In practice, we would use much larger p and q

