MA/CSSE 473 Day 10

Primality testing summary

Data Encryption

RSA

MA/CSSE 473 Day 10

- Student questions?
- Next Session, come prepared to discuss the interview with Donald Knuth (read it if you have not already done so – linked from schedule page, Session 3)
  - and Brute Force Algorithms
  - - and amortization
- Today:
  - Cryptography Introduction (Section 2)
  - RSA

We'll only scratch the surface, but there is MA/CSSE 479

CRYPTOGRAPHY INTRODUCTION

Cryptography Scenario

- I want to transmit a message **m** to you
  - in a form **e**(**m**) that you can readily decode by running **d**(**e**(**m**)),
  - And that 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.
  - Then I can use e to encode messages that only you* can decode

* and anyone else who can figure out what d is if they know e.

Messages can be integers

- Since a message is a sequence of bits …
- We can consider the message to be a sequence of b-bit integers (where b is fairly large), and encode each of those integers.
- Here we focus on encoding and decoding a single integer.

RSA Public-key Cryptography

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

bijection: a function f from a set X to a set Y with the property that for every y in Y, there is exactly one x in X such that f(x) = y. In other words, f is both one-to-one and onto.

N = p q

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

Q3-4

Public and Private Keys

- The first (bijection) property tells us that $x \rightarrow x^e \bmod N$ 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
  - When you receive a message m', you can decode it by calculating $(m')^d \bmod N$.

Example (from Wikipedia)

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

Q5-6

Recap: RSA Public-key Cryptography

- Consider a message to be a number modulo N, n k-bit number (longer messages can be broken up into n-bit pieces)
- 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 $x \rightarrow x^e$ mod N is a bijection on {0, 1, …, N-1}
  - If d is the inverse of e mod (p-1)(q-1), then for all x in {0, 1, …, N-1}, $(x^e)^d \equiv x$ (mod N).
- We have applied the property, now we prove it

RSA security

- **Assumption** (Factoring is hard!):
  - Given N, e, and x^e mod N, it is computationally intractable to determine x
  - What would it take to determine x?
- Presumably this will always be true if we choose N large enough
- But people have found other ways to attack RSA, by gathering additional information
- So these days, more sophisticated techniques are needed.
- MA/CSSE 479

Student questions

- On primality testing, RSA or anything else?