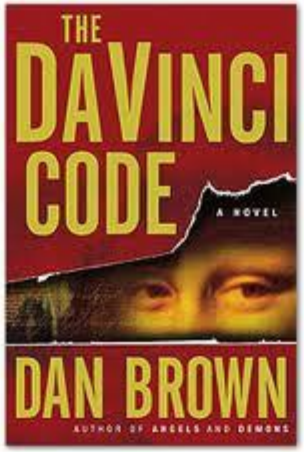
Exploration Session 4: RSA / Cryptography

Melissa Winstanley
mwinst@cs.washington.edu

(based on slides by Stuart Reges, Daniel Halperin)



Cryptography

- The science of keeping your information safe
- Only one small bit of larger system
  - Physical security
  - Operating system security
  - Network security
  - Users
  - Cryptography
- “Security only as strong as the weakest link”

Some Terminology

- **Alice** wants to send a secret message to **Bob**
- **Eve** is eavesdropping
- **Cryptographers** tell Alice and Bob how to encode their messages
- **Cryptanalysts** help Eve to break the code
- Historic **battle** between the cryptographers and the cryptanalysts that continues today

Private-Key Cryptography

- "Traditional" method of encryption
- Encrypt using a function of a key k
  - Alice encrypts with $c = (p + k) \bmod 26$
  - This is called a shift cipher
- Decrypt using the inverse function of key k
  - Bob decrypts with $p = (c - k) \bmod 26$
- Anyone who knows the key can read the message
- Alice and Bob must transmit the key securely

Public-Key Cryptography

- Proposed by Diffie, Hellman, Merkle
- First big idea: use a function that *cannot be reversed* (a humpty dumpty function)
  - Bob tells Alice a function to apply using a public key, and Eve can't compute the inverse
- Second big idea: use asymmetric keys (sender and receiver use different keys)
  - Bob has a private key to compute the inverse
- Primary benefit: doesn't require the sharing of a secret key

RSA Encryption

- Named for Ron Rivest, Adi Shamir, and Leonard Adleman
- Invented in 1977, still the premier approach
- Requires large primes (100+ digit primes)
- Effective because there is no current way to factor large primes

Review of mod

Basis of RSA

- Based on Fermat's Little Theorem:

  $a^{p-1} \equiv 1 \pmod{p}$ for prime p

  → gcd(a, p) = 1

- Slight variation:

  $a^{(p-1)(q-1)} \equiv 1 \pmod{pq}$ for distinct primes p, q

  → gcd(a,pq) = 1

Example of RSA

- Pick two primes p and q, compute $n = p \times q$
- Pick two numbers e and d, such that:

  $e \times d = (p-1)(q-1)k + 1$ (for some k)
- Publish n and e (public key), encode with:

  $(\text{original message})^e \bmod n$
- Keep d, p and q secret (private key), decode with:

  $(\text{encoded message})^d \bmod n$

Why does it work?

- Original message is carried to the e power, then to the d power:
  $$(msg^e)^d = msg^{ed}$$
- Remember how we picked e and d:
  $$msg^{ed} = msg^{(p-1)(q-1)k + 1}$$
- Apply some simple algebra:
  $$msg^{ed} = (msg^{(p-1)(q-1)})^k \times msg^1$$
- Applying Fermat's Little Theorem:
  $$msg^{ed} = (1)^k \times msg^1 = msg$$

Politics

- British discovered RSA first but kept it secret
- Phil Zimmerman tried to bring cryptography to the masses w/PGP
  - Investigated as an arms dealer by FBI and a grand jury
- Shor’s algorithm would break RSA if only we had a quantum computer
- The NSA hires more mathematicians than any other organization

http://www.nsa.gov/kids/