CS 113
DISCRETE STRUCTURES

Chapter 3: Algorithms

HOMEWORK

- **All homework is from the Exercises**
  - **No problems are from the Review Exercises**
- **Section 3.2 (p. 127): 19, 21**
- **Section 3.3 (p. 131): 1-5 (all), 13, 22-24 (all)**
- **Section 3.4 (p. 137): 8, 9, 12, 27, 28**
- **Section 3.5 (p. 149): 1-12 (all), 17, 19**
- **Section 3.7 (p. 160): 1-9 (all)**

THE RSA PUBLIC KEY CRYPTOSYSTEM

- **We saw some of these ideas in the chapter in the C++ textbook that talked about files**
  - **The textbook had examples of the Caesar cipher**
- **Cryptology is the study of ways of encoding data**
- **This is especially important in computer science**
  - **It's useful for security**
  - **Similar techniques are used for sending data over a noisy channel**

VOCABULARY

- **Using a formula to convert your raw data to something unreadable is called *encrypting* the data**
- **Using a formula to convert your encrypted data back to the original message is called *decrypting* the data**
- **Your original message is called *plaintext***
- **A message that is encrypted is called *ciphertext***

ENCRYPTING DATA

- One way to encrypt data is to use keys
- The person with plaintext uses a key to encrypt the data
  - You can think of a key as a number to add to or subtract from, each character of the plaintext
  - Or, you can think of a key as a number to multiply each character of the plaintext before adding them, etc.
- Then the message is sent
- The receiver also has a key
  - The key is used to decode the message

A PROBLEM

- One problem that we have is this:
  - How can the sender get the key to the receiver?
- The sender cannot just send it unencrypted
  - If the sender does that, anyone can see the key and the messages will not be secret any more
- But, there is no way yet to encrypt the key because the receiver won't know how to decrypt the encrypted key
- This is a problem!
- We look at this from the receiver's point of view
  - The receiver needs to decrypt the message, and so comes up with the encryption key

THE RSA ALGORITHM

- **The name comes from the three people who invented the algorithm**
  - **Ronald Rivest, Adi Shamir, Leonard Adelman**
- **The idea of the algorithm is simple**
- **There are two keys, an encryption key (to encrypt plaintext) and a decryption key (to decrypt ciphertext)**

THE KEYS

- **The receiver broadcasts the encryption key**
  - **It's now public**
  - **Anyone can see it and use it**
- **The receiver keeps the decryption key**
- **You might think having the encryption key also gives you the decryption key**
  - **It's probably just some math formula to encrypt, after all**
  - **How hard can it be to decrypt an encrypted message?**

THE ALGORITHM

- The algorithm depends on two numbers z and n
- z is chosen to be the product of two primes p and q
  - p and q are chosen to have a lot of digits, at least 100
  - For my example, I will use p = 3 and q = 5
  - So z = 15
- The receiver calculates $\phi = (p - 1)(q - 1)$
  - Here $\phi = (3 - 1)(5 - 1) = 2(4) = 8$
- The receiver chooses a number n with gcd $(n, \phi) = 1$
  - This is often chosen to be a prime
  - I will choose n = 11
- The receiver then computes s where $0 < s < \phi$ with $ns \bmod \phi = 1$
  - I will choose s = 3
- The receiver then publicly broadcasts z and n

ENCRYPTING THE CIPHERTEXT

- The receiver broadcasts z = 15, and n = 11
- Suppose the message is 3, 8
- It is encrypted as $c = a^n \bmod z$, which is $c = 3^{11} \bmod 15$
- $3^{11} = (3^4)^2(3^3) = (81)^2(3^3) \equiv (6^2)(3^3) \equiv 6(3^3) = 6(27)$
  $\equiv 6(12) = 72 \equiv 12 \bmod 15$
- $8^{11} = (8^2)^5(8) = (64)^5(8) \equiv (4)^5(8) = (4^2)^2x4(8) = (4^2)^2(4x8)$
  $= (16)^2x32 \equiv (1)^2x32 = 32 \equiv 2 \bmod 15$
- So, the message 12, 2 is sent

DECRYPTING THE CIPHERTEXT

- **The receiver sees the message and computes c^s mod z, which is c^3 mod 15**
- **$12^3 = (12^2)(12) \equiv 9 \times 12 = 108 \equiv 3$ mod 15**
- **$2^3 = 8$ mod 15**
- **So the decoded message is 3, 8 as it should be**
- **For future reference**
  - **z and n are the public keys**
  - **s is the private key**

INTERNET INTERLOPERS

- **Suppose you wanted to spy on the sender and the receiver**
- **You know z (which is 15) and n (which is 11)**
  - **They were broadcast publicly**
- **You see 12 flying by, then 2**
- **You have to decode 12^s mod 15 and 2^s mod 15, without knowing s**
  - **You do, however, know that $ns \equiv 1 \bmod \phi$, or $11s \equiv 1 \bmod \phi$**
- **Great! Except you don't know ϕ either**
- **☹ or, "That message is secure"**

CHECKING THE ALGORITHM

- **Why does this work?**
- **If a is the data to be encrypted, then $c = a^n \bmod z$ is sent**
  - **We know that this means that $c = a^n + vz$ for some integer v**
- **We need to recover a from c**
- **To start, we calculate c^s, which is $c^s = (a^n + vz)^s$**
- **Let's expand this by the binomial theorem**

FERMAT'S LITTLE THEOREM

- **Fermat was a mathematician who lived in the 1600s**
  - **He is famous for his "Last Theorem"**
- **I will rely on his little theorem**
- **It says that**

  If p is a prime, then $a^p \equiv a \pmod{p}$
- **This is equivalent to $a^{p-1} \equiv 1 \pmod{p}$**
  - **I divided both sides by a**
- **I will need this for later**

A SIMPLIFICATION USING MODULAR ARITHMETIC

- **Remember that $ns \bmod \phi = 1$**
  - **This means that $ns - 1 = u\phi$ for some integer u**
- **So then, $a^{ns} = a^{u\phi + 1} = (a^{u\phi})a = a^{u(p-1)(q-1)}a$**

CHECKING THE ALGORITHM-PART 2

- **Expanding $(a^n + vz)^s$ by the Binomial Theorem gives**
  - **$c^s = (a^n + vz)^s = a^{ns} + C_{s,1}a^{n(s-1)}(vz)^1 + C_{s,2}a^{n(s-2)}(vz)^2 + \ldots$**
- **$+ C_{s,s-1}a^{n(1)}(vz)^{s-1} + C_{s,0}(vz)^s$**
- **$= a^{ns} + (vz)[$ ---------$]$**
  - **Notice that vz is a factor of all terms except the first**
  - **Also, notice that $vz \equiv 0 \bmod z$**
- **So, $c^s \equiv a^{ns} \bmod z$, and by Fermat's Little Theorem,**
- **$a^{ns} = a^{u(p-1)(q-1)}a \equiv [a^{(p-1)}]^{(q-1)u}\, a \equiv 1^{(q-1)u}\, a \bmod z$**

  $\equiv a \bmod z$

HOMEWORK

- **Do the homework at the end of Section 3.7**
- **Section 3.7 (p. 160): 1-9 (all)**

QUESTIONS

- Any questions?