

CSE 20—Discrete Math

Fall, 2005

October 13 (Day 7)
Cryptography and Secrecy

Instructor: Neil Rhodes

Modulo

Remainder when dividing two positive integers (mod as an operator):

$$n = m \cdot \lfloor n/m \rfloor + (n \bmod m)$$

Extended to negative numbers ($y \neq 0$)

$$x \bmod y = x - y \cdot \lfloor x/y \rfloor$$

Equivalence classes (mod as an equivalence relation)

$$a \equiv b \pmod{m} \text{ iff } a-b \text{ is a multiple of } m$$

Residue classes mod m

- m of them
 - All a such that $a \equiv 0 \pmod{m}$
 - All a such that $a \equiv 1 \pmod{m}$
 - ...
 - All a such that $a \equiv m-1 \pmod{m}$

2

Modulo Arithmetic

If:

- $x = x' \pmod{m}$
- $y = y' \pmod{m}$

Then

- $x + y \equiv x' + y' \pmod{m}$
- $x - y \equiv x' - y' \pmod{m}$
- $xy \equiv x'y' \pmod{m}$

Applications

- Casting out 9's

$$\begin{array}{r} 532 \\ + 656 \\ \hline 1273 \end{array} \qquad \begin{array}{r} 327 \\ \times 55 \\ \hline 17985 \end{array}$$

3

gcd, lcm, Φ

Greatest Common Divisor (gcd)

- $\text{gcd}(m, n)$ is the largest integer k that divides integers m and n
 - $k \mid m$ and $k \mid n$
- $\text{gcd}(m, n)$ is a *linear combination* (with integer coefficients) of m and n
 - $\exists i, j \in \mathbb{Z}: \text{gcd}(m, n) = im + jn$
- To calculate $\text{gcd}(m, n)$
 - Compute prime factorization of m and n
 - $\text{gcd}(m, n)$ = common prime factors (and powers) of m and n
- Euclid's algorithm
 - $\text{int gcd}(m, n)$
 - if $(n == 0)$ return m
 - else return $\text{gcd}(n, m \% m)$

Least Common Multiple (lcm)

- $\text{lcm}(m, n)$ is the smallest integer k such that integers m and n divide k
 - $m \mid k$ and $n \mid k$
- To calculate $\text{lcm}(m, n)$
 - Compute prime factorizations of m and n
 - $\text{lcm}(m, n)$ = union of prime factors (and powers) of m and n

Φ (Euler function)

- $\Phi(n)$ = the number of positive integers $k \leq n$ such that $k \perp n$ ($\text{gcd}(k, n) = 1$)

4

Properties of Φ

If n is prime

- $\Phi(n) = n - 1$

If n is product of two distinct primes, p and q

- $\Phi(n) = (p-1)(q-1)$
 - Number not relatively prime = $q + p - 1$.
 - $pq - (q + p - 1) = (p-1)(q-1)$

Call the $\Phi(n)$ numbers relatively prime to n units

- Given a unit of n , a : $a^{\Phi(n)} \equiv 1 \pmod{n}$

Thus, given two distinct primes, p and q

- $m \perp pq \rightarrow m^{(p-1)(q-1)} \equiv 1 \pmod{pq}$

5

Cryptography

Definitions:

- Plaintext: message being encoded
- Ciphertext: encoded plaintext
- Key: parameter to crypto algorithms
 - $C = E(P, K_E)$
 - $P = D(C, K_D)$

Simple cipher (Caesar cipher):

- $K_E = K_D$ = permutation from letters to letters
 - $A \rightarrow X$
 - $B \rightarrow L$
 - ...
 - $Z \rightarrow R$
- E = apply K_E to each character in plaintext
- D = apply K_E^{-1} to each character in ciphertext
- Weakness: letter frequencies

6

Unbreakable Code

One-time pad

- $K_E = K_D$ = long stream of random bytes
 - Really random, not pseudo-random
- $E(P, K_E)$
 - for $i = 1$ to $\text{length}(P)$
 $C[i] = P[i] \text{ XOR } K_E[i]$
- $D = E$
- Alice sends message to Bob
 - Uses secret one-time pad
 - Encrypts P
 - Destroys P and one-time pad
- Bob decrypts
 - Using one-time pad
- Alternative encryption (by-hand)
 - Modular arithmetic
- Weakness
 - Key must be as long as plaintext
 - Key must be used only once
 - Key must be truly random

7

Reusing Key

Problem

- If plaintext and ciphertext are both known, key can be reverse-engineered
 - $K_E[i] = P[i] \text{ XOR } C[i]$
- How to know plaintext if it is encrypted?
 - Cause specific plaintext to be sent
 - British would mine specific areas in WWII so that the Germans would send message including "minen" and location.

8

Trapdoor Functions

Trapdoor function: Given the output of a function, the input is hard to compute

Discrete logarithms

- Consider computing a power of a number modulo a prime
 - $3^t \bmod 7$
 - Easy to compute
- Consider the reverse problem, given $3^t \bmod 7$, what is t ?
 - Believed to be difficult to compute for large numbers
 - t is the discrete logarithm

t	$3^t \bmod 7$
0	
1	
2	
3	
4	
5	
6	

Can be used to re-use keys

- Instead of encrypting with K , pick random b and encrypt with $L = b^K \bmod p$ (known prime p)
- Send C as well as b
- If attacker knows plaintext, can figure out L , but that doesn't help figure out K

9

Cryptography without a Shared Key

Symmetric Key Exchange (Diffie-Hellman)

- Alice and Bob don't have a shared key, but want to exchange information such that they know a shared key but attackers don't.
- Alice and Bob agree on prime p and base z ($1 < x < p-1$)
 - Standard numbers, perhaps
- Alice picks a random number a ($1 < a < p-1$), computes $A = z^a \bmod p$
- Bob picks a random number b ($1 < b < p-1$), computes $B = z^b \bmod p$
- Alice and Bob exchange A and B
 - Everyone knows A , B , p , and z
- Alice computes $K = B^a \bmod p$
- Bob computes $K = A^b \bmod p$
 - $B^a \equiv (z^b)^a \equiv z^{(ba)} \equiv z^{(ab)} \equiv (z^a)^b \equiv A^b \pmod{p}$

10

Cryptography without a Shared Key

Public Key Cryptography (RSA)

- Alice has a public key, K_E , used for encrypting messages to her
- She has a secret private key, K_D , used for decryption
- Creating pair of keys:
 - Choose two large random primes p , and q
 - Compute $n = pq$
 - Compute $\Phi(n) = (p-1)(q-1)$
 - Choose an integer $1 < e < \Phi(n)$ such that $e \perp \Phi(n)$
 - Compute d such that $de \equiv 1 \pmod{\Phi(n)}$
- $K_E = (n, e)$, $K_D = (n, d)$
- Choose P with $1 \leq P < n$, and $P \perp n$
- Encrypt: $C = P^e \bmod n$
- Decrypt: $P = C^d \bmod n$
 - $C^d \equiv (P^e)^d \equiv P^{ed} \pmod{n}$
 - Since $de \equiv 1 \pmod{\Phi(n)}$, $de = 1 + i\Phi(n)$
 - So, $P^{ed} \equiv P^{1+i\Phi(n)} \equiv P^{1+i\Phi(n)} \equiv P \cdot P^{i\Phi(n)} \pmod{n}$
 - But, $P^{i\Phi(n)} \equiv 1 \pmod{n}$, so $C^d \equiv P \pmod{n}$
- Weakness/strength:
 - If n can be factored, private key can be easily recreated

11