Introduction to Cryptography
Lecture 5

Basic Number Theory

Benny Pinkas

Plan

- Today
  - Basic number theory
    - Divisors, modular arithmetic
    - The GCD algorithm
    - Groups
  - References:
    - Many book on number theory
    - Almost all books on cryptography
    - Cormen, Leiserson, Rivest, (Stein), “Introduction to Algorithms”, chapter on Number-Theoretic Algorithms.

Divisors, prime numbers

- We work over the integers
- A non-zero integer *b* divides an integer *a* if there exists an integer *c* s.t. $a=c\cdot b$.
  - Denoted as $b|a$
  - I.e. *b* divides *a* with no remainder
- Examples
  - Trivial divisors: $1|a$, $a|a$
  - Each of {1,2,3,4,6,8,12,24} divides 24
  - 5 does not divide 24
- Prime numbers
  - An integer *a* is prime if it is only divided by 1 and by itself.
  - 23 is prime, 24 is not.

Modular Arithmetic

- Modular operator:
  - *a* mod *b*, (or *a%b*) is the remainder of *a* when divided by *b*
  - I.e., the smallest $r \geq 0$ s.t. $\exists$ integer *q* for which $a = qb+r$.
  - (Thm: there is a single choice for such *q*,*r*)

  - Examples
    - *12 mod 5 = 2*
    - 10 mod 5 = 0
    - -5 mod 5 = 0
    - -1 mod 5 = 4

Modular congruency

- *a* is congruent to *b* modulo *n* ($a \equiv b$ mod *n*) if
  - $(a-b) = 0$ mod *n*
  - Namely, *n* divides *a-b*
  - In other words, (*a* mod *n*) = (*b* mod *n*)
- E.g.,
  - $23 \equiv 12$ mod 11
  - $4 \equiv -1$ mod 5
- There are *n* equivalence classes modulo *n*
  - $[3]_7 = \{\ldots,-11,-4,3,10,17,\ldots\}$

Greatest Common Divisor (GCD)

- *d* is a common divisor of *a* and *b*, if $d|a$ and $d|b$.
- *gcd(a,b)* (Greatest Common Divisor), is the largest integer that divides both *a* and *b*. ($a,b >= 0$)
  - $gcd(a,b)$ = max *k s.t.* $k|a$ and $k|b$.
- Examples:
  - gcd(30,24) = 6
  - gcd(30,23) = 1
- If $gcd(a,b)=1$ they are denoted relatively prime.

Facts about the GCD

- $gcd(a,b) = gcd(b, a \text{ mod } b)$ (interesting when a>b)
- Since (e.g., a=33, b=15)
  - If $c|a$ *and* $c|b$ then $c|(a \text{ mod } b)$
  - If $c|b$ and $c|(a \text{ mod } b)$ then $c|a$
- If *a* mod *b* = 0, then gcd(a,b)=b.
- Therefore,

gcd(19,8) =
gcd(8, 3) =
gcd(3,2) =
gcd(2,1) = 1

gcd(20,8) =
gcd(8, 4) = 4

Euclid's algorithm

Input: a>b>0
Output: gcd(a,b)
Algorithm:

1. if (*a mod b*) = *0* return (*b*)
2. else return(gcd(*b*, *a* mod *b*))

Complexity:

- O(log *a*) rounds
- Each round of overhead $O(\log^2 a)$ bit operations
- Actually, the total overhead can be shown to be $O(\log^2 a)$

The extended gcd algorithm

Finding *s, t* such that $gcd(a,b) = as+bt$

Extended-gcd(a,b) /* output is (gcd(a,b), s, t)

1. If (a mod b=0) then return(b,0,1)
2. (d',s',t') = Extended-gcd(b, a mod b)
3. (d,s,t) = (d', t', s'-$\lfloor a/b \rfloor \cdot$t')
4. return(d,s,t)

Note that the overhead is as in the basic GCD algorithm

Groups

- Definition: a set G with a binary operation $\circ$:G$\times$G$\rightarrow$G is called a group if:
  - (closure) $\forall a,b \in G$, it holds that $a \circ b \in G$.
  - (associativity) $\forall a,b,c \in G$, $(a \circ b) \circ c = a \circ (b \circ c)$.
  - (identity element) $\exists e \in G$, s.t. $\forall a \in G$ it holds that $a \circ e = a$.
  - (inverse element) $\forall a \in G\ \exists a^{-1} \in G$, s.t. $a \circ a^{-1} = e$.
- A group is Abelian (commutative) if $\forall a,b \in G$, it holds that $a \circ b = b \circ a$.
- Examples:
  - Integers under addition
    - (Z,+) = {…,-3,-2,-1,0,1,2,3,…}

More examples of groups

- Addition modulo N
  - $(G, \circ) = (\{0,1,2,\ldots,N-1\}, +)$
- Z_p^* Multiplication modulo a prime number *p*
  - $(G, \circ) = (\{1,2,\ldots,p-1\}, \times)$
  - E.g., $Z_7^* = (\{1,2,3,4,5,6\}, \times)$
- Trivial: closure (the result of the multiplication is never divisible by *p*), associativity, existence of identity element.
- The extended GCD algorithm shows that an inverse always exists:
  - $s \cdot a + t \cdot p = 1 \Rightarrow s \cdot a = 1 - t \cdot p \Rightarrow s \cdot a \equiv 1 \mod p$

More examples of groups

- Z_N^* Multiplication modulo a composite number *N*
  - $(G, \circ) = (\{a$ s.t. $1 \le a \le N-1$ and $gcd(a,N)=1\}, \times)$
  - E.g., $Z_{10}^* = (\{1,3,7,9\}, \times)$
  - Closure:
    - $s \cdot a + t \cdot N = 1$
    - $s' \cdot b + t' \cdot N = 1$
    - $ss' \cdot (ab) + (sat' + s'bt + tt'N) \cdot N = 1$
  - Associativity: trivial
  - Existence of identity element: 1.
  - Inverse element: as in Z_p^*

Subgroups

- Let $(G,\circ)$ be a group.
  - $(H,\circ)$ is a subgroup of G if
    - $(H,\circ)$ is a group
    - $H \subseteq G$
  - For example, $H = (\{1,2,4\}, \times)$ is a subgroup of Z_7^*.
- *Lagrange's theorem:*

  If $(G,\circ)$ is finite and $(H,\circ)$ is a subgroup of $(G,\circ)$, then $|H|$ divides $|G|$

  For example: 3|6.

Cyclic Groups

- Exponentiation is repeated application of $\circ$
  - $a^3 = a \circ a \circ a.$
  - $a^0 = 1.$
  - $a^{-x} = (a^{-1})^x$
- A group G is cyclic if there exists a generator g, s.t. $\forall a \in G,\ \exists i$ s.t. $g^i = a$.
  - I.e., $G = \langle g \rangle = \{1, g, g^2, g^3, \ldots\}$
  - For example $Z_7^* = \langle 3 \rangle = \{1,3,2,6,4,5\}$
- Not all $a \in G$ are generators of G, but they all generate a subgroup of G.
  - E.g. 2 is not a generator of Z_7^*
- The order of a is the smallest $j>0$ s.t. $a^j = 1.$
- *Lagrange's theorem* $\Rightarrow$ for $x \in Z_p^*$, $ord(x) \mid p-1.$

Fermat's theorem

- Corollary of Lagrange's theorem: if $(G,\circ)$ is a finite group, then $\forall a \in G,\ a^{|G|} = 1.$
- Corollary (Fermat's theorem): $\forall a \in Z_p^*,\ a^{p-1} = 1 \bmod p.$ E.g., for all $\forall a \in Z_7^*$, $a^6 = 1,\ a^7 = a.$
- Computing inverses:
- Given $a \in G$, how to compute a^{-1}?
  - Fermat's theorem: $a^{-1} = a^{|G|-1}$ $(= a^{p-2}$ in $Z_p^*)$
  - Or, using the extended gcd algorithm (for Z_p^* or Z_N^*):
    - $gcd(a,p) = 1$
    - $s \cdot a + t \cdot p = 1 \Rightarrow s \cdot a = -t \cdot p + 1 \Rightarrow$ s is a^{-1} !!
  - Which is more efficient?

Computing in Z_p^*

- P is a huge prime (1024 bits)
- Easy tasks (measured in bit operations):
  - Adding in O(log p) (linear n the length of p)
  - Multiplying in O($\log^2$ p) (and even in O($\log^{1.7}$ p))
  - Inverting (a to a^{-1}) in O($\log^2$ p)
  - Exponentiations:
    - $x^r \bmod p$ in O(log r · $\log^2$ p), using repeated squaring