Introduction to Cryptography
Lecture 4

Basic Number Theory
Diffie-Hellman Key Exchange

Benny Pinkas

Plan

- Today
  - Basic number theory
    - Divisors, modular arithmetic, gcd
    - Groups
  - Public key cryptography
    - Diffie-Hellman key exchange

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
  - $6|12$
  - 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 $a = qb+r$.
  - Or, $0 \leq r < b$, s.t. $a = qb + r$.
  - Examples
    - $7 \bmod 5 = 2$
    - 10 mod 5 = 0
    - -5 mod 5 = 0
    - -1 mod 5 = 4

Modular congruency

- a is congruent to b modulo n $(a \equiv b \bmod n)$ if
  - $(a-b) = 0 \bmod n$
  - I.e., $(a \bmod n) = (b \bmod n)$
- E.g.,
  - $23 \equiv 12 \bmod 11$
  - $4 \equiv -1 \bmod 5$

Greatest Common Divisor (GCD)

- *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* mod *b)* (interesting when a>b)
- Since
  - If *c|a and c|b* then *c|(a mod b)*
  - If *c|b* and *c|(a 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)

Alg:

1. if ($a \bmod b) = 0$ return (b)
2. else return(gcd(b, a mod b))

Complexity:

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

The extended gcd algorithm

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

$R_0 = a$		$S_0 = 1$	$T_0 = 0$
$R_1 = b$	$Q_1 = \lfloor R_0 / R_1 \rfloor$	$S_1 = 0$	$T_1 = 1$
$R_2 = R_0 \bmod R_1$	$Q_2 = \lfloor R_1 / R_2 \rfloor$	$S_2 = S_0 - Q_1 S_1$	$T_2 = T_0 - Q_1 T_1$
$R_i = R_{i-2} \bmod R_{i-1}$	$Q_i = \lfloor R_{i-1} / R_i \rfloor$	$S_i = S_{i-2} - Q_{i-1} S_{i-1}$	$T_i = T_{i-2} - Q_{i-1} T_{i-1}$

It holds that $R_i = a \cdot S_i + b \cdot T_i$ (by induction)

Therefore, $gcd(a,b) = R_n = a \cdot S_n + b \cdot T_n$

same overhead as computing gcd

Groups

- Definition: a set G with a binary operation $\circ$:G×G→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., $Z_7^* = (\{1,2,3,4,5,6\}, \times)$

- Z_N^* Multiplication modulo a composite number N
  - $(G, \circ) = (\{a \text{ s.t. } 1 \leq a \leq N-1 \text{ and } gcd(a,N)=1\}, \times)$
  - E.g., $Z_{10}^* = (\{1,3,7,9\}, \times)$

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:
    - $gcd(a,p) = 1$
    - $s \cdot a + t \cdot p = 1 \Rightarrow s \cdot a = -t \cdot p + 1 \Rightarrow s$ is a^{-1} !!
  - Which is more efficient?

Computing in Z_p^*

- P is a huge prime (1024 bits)
- Easy tasks:
  - 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 $O(\log^2 p)$
  - Solving linear equations? $ax+b = x \bmod p$
  - Exponentiations:
    - $x^r \bmod p$ in $O(\log r \cdot \log^2 p)$, using repeated squaring

Hard problems in Z_p^*

- The following problems are believed to be hard
- Discrete logarithm: let g be a generator of Z_p^*. Given a random $x \in Z_p^*$ find an r such that $x = g^r \bmod p$.
- The Diffie-Hellman problem: Given random $x, y \in Z_p^*$, such that $x = g^a$ and $y = g^b$, *find* $z = g^{a \cdot b}$.
- The Decisional Diffie-Hellman problem: Given random $x, y \in Z_p^*$, such that $x = g^a$ and $y = g^b$; *and a value* z which is promised to be either $g^{a \cdot b}$ *or* g^c (for a random c), tell which is the case.
- DL > DH > DDH

Classical symmetric ciphers

- Alice and Bob share a private key *k.*
- System is secure as long as *k* is secret.
- Major problem: generating and distributing *k.*

Diffie and Hellman: “New Directions in Cryptography”, 1976.

- “We stand today on the brink of a revolution in cryptography. The development of cheap digital hardware has freed it from the design limitations of mechanical computing…

  …such applications create a need for new types of cryptographic systems which minimize the necessity of secure key distribution…

  …theoretical developments in information theory and computer science show promise of providing provably secure cryptosystems, changing this ancient art into a science.”

Diffie-Hellman

- Came up with the idea of public key cryptography

Everyone can learn Bob's public key and encrypt messages to Bob.
Only Bob knows the decryption key and can decrypt.

Key distribution is greatly simplified.

- Diffie and Hellman did not have an implementation for a public key encryption system
- Suggested a method for key exchange over insecure communication lines, that is still in use today.

Public Key-Exchange

- Goal: Two parties who do not share any secret information, perform a protocol and derive the same shared key.
- No eavesdropper can obtain the new shared key (if it has limited computational resources).
- The parties can therefore safely use the key as an encryption key.

The Diffie-Hellman Key Exchange Protocol

- Public parameters: a group Zp^* (where $|p| = 768$ or 1024), and a generator g.

- Alice:
  - picks a random $a \in [1,p-1]$.
  - Sends g^a mod p to Bob.
  - Computes $k=(g^b)^a$ mod p

- Bob:
  - picks a random $b \in [1,p-1]$.
  - Sends g^b mod p to Bob.
  - Computes $k=(g^a)^b$ mod p

- $K = g^{ab}$ is used as a shared key between Alice and Bob.

Diffie-Hellamn: security

- A *(passive)* adversary
  - Knows Z_p^*, g
  - Sees g^a, g^b
  - Wants to compute g^{ab}, or at least learn something about it
- Recall the Decisional Diffie-Hellman problem:
  - Given random $x,y \in Z_p^*$, such that $x=g^a$ and $y=g^b$*; and a value z* which is promised to be either g^{ab} *or* g^c (for a random c), it is hard tell which is the case.
  - I.e., g^{ab} is indistinguishable from a random element in Z_p^*.
  - *Note:* it is insufficient to require that the adversary cannot compute g^{ab}.

Diffie-Hellman key exchange: usage

- The DH key exchange can be used in any group *in which the Decisional Diffie-Hellman (DDH) assumption is believed to hold.*
- Currently, Z_p^* and elliptic curve groups.
- Example: the additive group modulo p?
- Common usage:
  - Overhead: 1-2 exponentiations
  - Usually,
    - A DH key exchange for generating a master key
    - Master key used to encrypt session keys
    - Session key is used to encrypt traffic with a symmetric cryptosystem