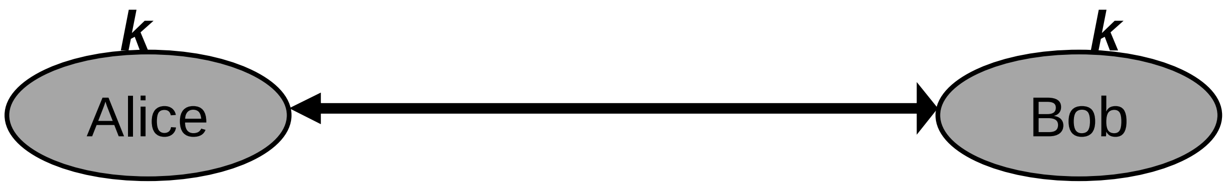
Topics in Cryptography

Lecture 5: Basic Number Theory

Benny Pinkas

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

public key$_{Bob}$ *secret key*$_{Bob}$

Alice Bob

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.

But before we get to public key cryptography…

- Basic number theory
  - Divisors, modular arithmetic
  - The GCD algorithm
  - Groups

- References:
  - Many books 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 divisible 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 $(a \equiv b \bmod n)$ if
  - $(a-b) = 0 \bmod n$
  - Namely, n divides $a-b$
  - In other words, $(a \bmod n) = (b \bmod n)$

- E.g.,
  - $23 \equiv 12 \bmod 11$
  - $4 \equiv -1 \bmod 5$

Modular congruency

- Modular congruency is an equivalence relation:
  - $\forall a$, ($a \equiv a$ mod n)
  - ($a \equiv b$ mod n) *implies* ($b \equiv a$ mod n)
  - ($a \equiv b$ mod n) *and* ($b \equiv c$ mod n) *imply* ($a \equiv c$ mod n)
  - There are n equivalence classes modulo n
    - $[3]_7 = \{\ldots,-11,-4,3,10,17,\ldots\}$

- If ($a \equiv a'$ mod n) *and* ($b \equiv b'$ mod n) then
  - $((a+b) \equiv (a'+b')$ mod $n)$
  - $((a \cdot b) \equiv (a' \cdot b')$ mod $n)$
  - *But* $((a \cdot b) \equiv (c \cdot b)$ mod $n)$ does not imply that ($a \equiv c$ mod n)
    - $3 \cdot 2 = 15 \cdot 2 = 6$ *mod 24.* But, ($3 \neq 15$ mod 24).

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 $gcd(a,b)=1$ then *a* and *b* are said to be relatively prime.

Facts about the GCD

- *gcd(a,b)* = *gcd(b, a* mod *b)* (interesting when a>b)
- Since (e.g., a=33, b=15)
  - 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)

Algorithm:

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

Complexity:

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

The extended gcd algorithm

Finding *s, t* such that $gcd(a,b) = a\cdot s + b\cdot t$

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

- Extended gcd algorithm
  - Given a,b finds s,t such that $\gcd(a,b) = a \cdot s + b \cdot t$
  - In particular, if p is prime than $\gcd(a,p)=1$, and therefore $a \cdot s+p \cdot t=1$. This implies that ($a \cdot s \equiv 1 \bmod p$)

- THM: There is no integer smaller than gcd(a,b) which can be represented as a linear combination of a,b.
  - For example, a=12, b=8.
  - $4= 1 \cdot 12 - 1 \cdot 8$
  - There are no s,t for which $2=s \cdot 12 + t \cdot 8$

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
    - $(Z,+) = \{\ldots,-3,-2,-1,0,1,2,3,\ldots\}$

More examples of groups

- Addition modulo N
  - $(G, \circ) = (\{0,1,2,\ldots,N\text{-}1\}, +)$

- Z_p^* Multiplication modulo a prime number p
  - $(G, \circ) = (\{1,2,\ldots,p\text{-}1\}, \times)$
  - 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:
  - $s \cdot a + t \cdot p = 1 \quad \Rightarrow \quad s \cdot a = 1 - t \cdot p \quad \Rightarrow \quad s \cdot a \equiv 1 \bmod p$

More examples of groups

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

  - Closure:
    - $s \cdot a + t \cdot N = 1$
    - $s' \cdot b + t' \cdot N = 1$
    - $ss' \cdot (ab) + (sat' + s'bt + tt'N) \cdot N = 1$
    - Therefore $1 = gcd(ab,N)$.
  - 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 $|G|$

  In our 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= <g> = {1, g, g^2, g^3, …}
  - For example $Z_7^* = <3> = \{1,3,2,6,4,5\}$
- 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 group element 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): $\forall\, a \in Z_p^*,\ \ a^{p-1} = 1$ mod $p.$ 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) (namely, 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 $O(\log r \cdot \log^2 p)$, using repeated squaring

Groups we will use

- Z_p^* Multiplication modulo a prime number p
  - $(G, \circ) = (\{1,2,\ldots,p\text{-}1\}, \times)$
  - E.g., $Z_7^* = (\{1,2,3,4,5,6\}, \times)$

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

Euler's phi function

- Lagrange's Theorem: $\forall a$ in a finite group G, $a^{|G|}=1$.
- Euler's phi function (aka, Euler's totient function),
  - $\phi(n)$ = number of elements in Z^*_n (i.e. $|\{x \mid gcd(x,n)=1, 1 \le x \le n\}|$
  - $\phi(p) = p-1$ for a prime p.
  - $n=\prod_{i=1..k} p_i^{e(i)} \Rightarrow \phi(n) = n \cdot \prod_{i=1..k} (1-1/p_i)$
  - $\phi(p^2) = p(p-1)$ for a prime p.
  - $n=p \cdot q \Rightarrow \phi(n) =(p-1)(q-1)$

- Corollary: For Z_n^* ($n=p \cdot q$), $|Z_n^*|= \phi(n) =(p-1)(q-1)$.
- $\forall a \in Z_n^*$ it holds that $a^{\phi(n)}=1$ mod n
  - For Z_p^* (prime p), $a^{p-1}=1$ mod p (Fermat's theorem).
  - For Z_n^* ($n=p \cdot q$), $a^{(p-1)(q-1)}=1$ mod n

Finding prime numbers

Finding prime numbers

- Prime number theorem: #{primes $\leq x$} $\approx x / \ln x$ as $x \to \infty$

- How can we find a random k-bit prime?
  - Choose x at random in $\{2^k, \ldots, 2^{k+1}-1\}$
    - (About $1 / \ln(2^k)$ of the numbers in that range are prime)
  - Test if x is prime
    - (more on this later in the course)
- The probability of success is $\approx 1/\ln(2^k) = O(1/k)$.
- The expected number of trials is $O(k)$.

Finding generators

- How can we find a generator of Z_p^*?
- Pick a random number $a \in [1,p-1]$, check if is a generator
  - Can check whether $\forall\, 1 \le i \le p-2 \quad a^i \ne 1$ ☹
  - We know that if $a^i = 1 \bmod p$ then $i \mid p-1$.
  - Therefore need to check only i for which $i \mid p-1$.

- Easy if we know the factorization of $(p-1)$
  - For all $a \in Z_p^*$, the order of a divides $(p-1)$
  - For every integer divisor b of $(p-1)$, check if $a^b=1$ mod p.
  - If none of these checks succeeds, then a is a generator.
  - a is a generator iff $ord(a)=p-1$.

Finding prime numbers of the right form

- How can we know the factorization of p-1
- Easy, for example, if $p=2q+1$, and q is prime.
- How can we find a k-bit prime of this form?

1. Search for a prime number q of length $k-1$ bits. (Will be successful after about $O(k)$ attempts.)
2. Check if $2q+1$ is prime (we will see how to do this later in the course).
3. If not, go to step 1.

Hard problems in cyclic groups

A hard problem can be useful for constructing cryptographic systems, if we can show that breaking the system is equivalent to solving this problem.

The Discrete Logarithm

- Let G be a cyclic group of order q, with a generator g.
  - $\forall h \in G,\ \exists x \in [1,\ldots,q]$, such that $g^x = h$.
  - This x is called the discrete logarithm of h to the base g.

  - $\log_g h = x$.
  - $\log_g 1 = 0$, and $\log_g(h_1 \cdot h_2) = \log_g(h_1) + \log_g(h_2) \bmod q$.

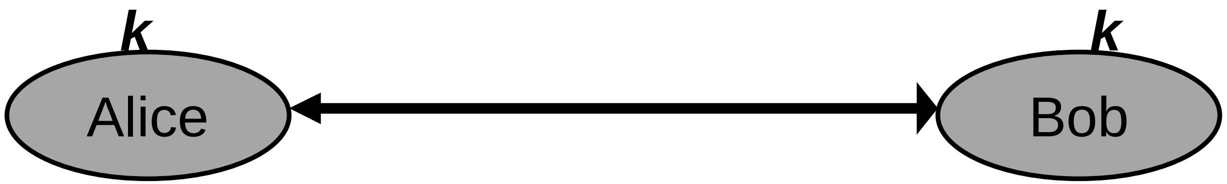
The Discrete Logarithm Problem and Assumption

- The discrete log problem
  - Choose *G,g* at random (from a certain family *G* of groups), where *G* is a cyclic group and *g* is a generator
  - Choose a random element $h \in G$
  - Give the adversary the input $(G,|G|,g,h)$
  - The adversary succeeds if it outputs $\log_g h$

- The discrete log assumption
  - There exists a family *G* of groups for which the discrete log problem is hard
    - Namely, the adversary has negligible success probability.

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 where the DDH assumption holds. For example, Z_p* (where $|p|= 768$ or $1024, p=2q+1$), and a generator g of $H\subset Z_p$* of order q.

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

- DDH assumption $\Rightarrow$ K is indistinguishable from a random key