Network Information Security

Lecture eight:
Basic number theory

1

Notation （标记）

From here on:

- N denotes a positive integer (正整数) .
- p denote a prime(素数).

Notation: $\mathbb{Z}_N$ = {0,1,2,…,N-1}

Can do addition and multiplication modulo N
(模N的加法与乘法)

2

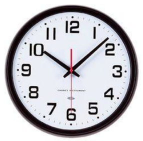
Modular arithmetic (模的算术运算)

Examples: let N = 12

$9 + 8 = \square$ in $\mathbb{Z}_{12}$

$5 \times 7 = \square$ in $\mathbb{Z}_{12}$

$5 - 7 = \square$ in $\mathbb{Z}_{12}$



还满足分配率等运算法则，如：$x \cdot (y+z) = x \cdot y + x \cdot z$ in $\mathbb{Z}_{12}$

3

Modular Arithmetic (模运算)

- Modular addition
  - [(a mod n) + (b mod n)] mod n = (a + b) mod n

  Example: [16 mod 12 + 8 mod 12] mod 12 = (16 + 8) mod 12 = 0
- Modular subtraction
  - [(a mod n) – (b mod n)] mod n = (a - b) mod n

  Example: [22 mod 12 - 8 mod 12] mod 12 = (22 - 8) mod 12 = 2
- Modular multiplication
  - [(a mod n) × (b mod n)] mod n = (a × b) mod n

  Example: [22 mod 12 × 8 mod 12] mod 12 = (22 × 8) mod 12 = 8

4

Properties of Modular Arithmetic

- Commutative laws (交换律)
  - $(w + x) \bmod n = (x + w) \bmod n$
  - $(w \times x) \bmod n = (x \times w) \bmod n$
- Associative laws (结合律)
  - $[(w + x) + y] \bmod n = [w + (x + y)] \bmod n$
  - $[(w \times x) \times y] \bmod n = [w \times (x \times y)] \bmod n$
- Distributive law (分配率)
  - $[w \times (x + y)] \bmod n = [(w \times x) + (w \times y)] \bmod n$

5

Greatest common divisor (最大公约数)

Def: For ints. x,y: **gcd(x, y)** is the greatest common divisor of x,y

Example: gcd(12, 18) = 6 $\boxed{2}\times 12 \boxed{-1}\times 18 = 6$

Fact: for all ints. x,y there exist ints. a,b such that

a·x + b·y = gcd(x,y)

a,b can be found efficiently using the extended Euclid alg (扩展的欧几里得算法).

If gcd(x,y)=1 we say that x and y are **relatively prime(互素)**

6

Greatest Common Divisor (最大公约数)

- Def: gcd(a,b) = max{k, such that k|a and k|b}
- Observations
  - gcd(a,b) = gcd(|a|, |b|)
  - gcd(a, 0) = |a|
  - gcd(a,b) ≤ min(|a|, |b|)
  - a and b are relatively prime if gcd(a, b) = 1
  - If 0 ≤ n, then gcd(an, bn) = n*gcd(a,b)

7

More properties of Common divisor

- A number d that is a divisor of both a and b is a common divisor of a and b

  Example: common divisors of 30 and 24 are 1, 2, 3, 6

- If d|a and d|b, then d|(a+b) and d|(a-b)

  Example: Since 3 | 30 and 3 | 24 , 3 | (30+24) and 3 | (30-24)

- If d|a and d|b, then d|(ax+by) for any integers x and y

  Example: 3 | 30 and 3 | 24 → 3 | (2*30 + 6*24)

8

How to compute GCD(x,y)?

9

Method one: 算术基本定理

算术基本定理

- Any integer a > 1 can be factored (因子分解) in a unique way as $p_1^{a1} \bullet p_2^{a2} \bullet \ldots p_t^{at}$
  - Where all $p_1 > p_2 \ldots > p_t$ are prime numbers and where each $a_i > 0$

Examples:
$91 = 13^1 \times 7^1$
$11011 = 13^1 \times 11^2 \times 7^1$

10

Finding the Greatest Common Divisor

Computing GCD by hand:
if $a = p_1^{a1} p_2^{a2} \ldots p_r^{ar}$ and
$b = p_1^{b1} p_2^{b2} \ldots p_r^{br}$,
…where $p1 < p2 < \ldots < pr$ are prime,
…and *ai* and *bi* are nonnegative,
…then $\gcd(a, b) =$
$p_1^{\min(a1, b1)} p_2^{\min(a2, b2)} \ldots p_r^{\min(ar, br)}$

11

Method two：Euclid's Algorithm for GCD

- Suppose we have integers x, y such that d = gcd(x, y). Because gcd(|x|, |y|) = gcd(x, y), assuming x ≥ y>0.

$$x = q_1 y + r_1 \qquad 0 \le r_1 < y$$

- **Prove:** $\gcd(x, y) = \gcd(y, r_1)$

辗转相除法

- Or gcd(x, y) = gcd(y, x mod y)

12

Euclid's Algorithm for GCD

$x = q_1 y + r_1 \quad 0<r_1<y$

$y = q_2 r_1 + r_2 \quad 0<r_2<r_1$

$r_1 = q_3 r_2 + r_3 \quad 0<r_3<r_2$

.

.

.

$r_{n-2} = q_n r_{n-1} + r_n \quad 0<r_n<r_{n-1}$

$r_{n-1} = q_{n+1} r_n + 0$

$d = \gcd(x,y) = r_n$

13

Procedure euclid(x, y)

```
r[0] = x, r[1] = y, n = 1;
while (r[n] != 0) {
   n = n+1;
   r[n] = r[n-2] % r[n-1];
}
return r[n-1];
```

14

Example

n	q_n	r_n
0	-	595
1	-	408
2	1	187
3	2	34
4	5	17
5	2	0

gcd(595,408) = 17

15

Exercise

Try to calculate GCD(1071,462) using Euclid algorithm.

Step k	Equation	Quotient and remainder
0	$1071 = q_0\, 462 + r_0$	$q_0 = 2$ and $r_0 = 147$
1	$462 = q_1\, 147 + r_1$	$q_1 = 3$ and $r_1 = 21$
2	$147 = q_2\, 21 + r_2$	$q_2 = 7$ and $r_2 = 0$; algorithm ends

16

Extended Euclid's Algorithm

- Let *LC*(*x*, *y*) = {*ax*+*by*: *a*,*b*∈*Z*} be the set of linear combinations (线性组合) of x and y.
- gcd(*x*,*y*) is the smallest positive value of *LC*(*x*, *y*).

ax + by = d = gcd (x, y)

- Euclid's algorithm can be extended to compute *a* and *b*, as well as gcd(*x*, *y*).
- Used in Multiplicative inverses and the RSA algorithm. (用在求模的逆以及RSA算法中)

17

Extended Euclid's Algorithm

floor function (向下取整)

```
r[0] = x, r[1] = y, n = 1;
u[0] = 1, u[1] = 0;
v[0] = 0, v[1] = 1;
while (r[n] != 0) {
  n = n+1;
  r[n] = r[n-2] % r[n-1];
  q[n] = (int) (r[n-2] / r[n-1]);
  u[n] = u[n-2] - q[n]*u[n-1];
  v[n] = v[n-2] - q[n]*v[n-1];
}
return r[n-1], u[n-1], v[n-1];
```

18

Extended Euclid's Example

n	q_n	r_n	u_n	v_n
0	-	595	1	0
1	-	408	0	1
2	1	187	1	-1
3	2	34	-2	3
4	5	17	11	-16
5	2	0	-24	35

gcd(595,408) = 17 = 11*595 + -16*408

19

Extended Euclid's Exercise

n	q_n	r_n	u_n	v_n
0	-	99	1	0
1	-	78	0	1
2	1	21	1	-1
3	3	15	-3	4
4	1	6	4	-5
5	2	3	-11	14
6		0		

20

Modular inversion (模的逆)

Def: The **inverse** of x in $\mathbb{Z}_N$ is an element y in $\mathbb{Z}_N$ s.t.

$$x \cdot y = 1 \ in \ \mathbb{Z}_N$$

y is denoted x^{-1} .

Example: let N be an odd integer. The inverse of 2 in $\mathbb{Z}_N$ is

$$\frac{N+1}{2}$$

$$2 \cdot \frac{N+1}{2} = N + 1 = 1 \text{ in } \mathbb{Z}_N$$

21

Modular inversion

Which elements have an inverse in $\mathbb{Z}_N$?

Lemma: x in $\mathbb{Z}_N$ has an inverse, if and only if gcd(x,N) = 1

Proof:

gcd(x,N)=1 ⇒ ∃ a,b: a·x + b·N = 1

$\Rightarrow$ $a \cdot x = 1 \ in \ \mathbb{Z}_N$

$\Rightarrow$ $x^{-1} = a \ in \ \mathbb{Z}_N$

22

Finding the Multiplicative Inverse

- Given x and N, how do you find x^{-1} mod N?
- Extended Euclid's Algorithm
- **exteuclid(x, N)**

23

Example

- 12^{-1}mod35

n	q_n	r_n	u_n	v_n
0	-	35	1	0
1	-	12	0	1
2	2	11	1	-2
3	1	1	-1	$\boxed{3}$
4	11	0	12	-35

gcd(35,12) = 1 = -1*35 + 3*12

12^{-1} mod 35 = **3** (i.e., 12*3 mod 35 = 1)

24

More notation

Def: $\mathbb{Z}_N^*$ = (set of invertible elements in $\mathbb{Z}_N$) =
= { $x \in \mathbb{Z}_N$: gcd(x,N) = 1 }

Examples:

1. for prime p, $\mathbb{Z}_p^* = \{1, 2, \ldots, p-1\}$
2. $\mathbb{Z}_{12}^* = \{1, 5, 7, 11\}$

25

Fermat's theorem (1640)

Thm: Let p be a prime

$$\forall\, x \in Z_p^* : \quad x^{p-1} = 1 \text{ in } Z_p$$

Example: p=5. $3^4 = 81 = 1$ in Z_5

So: $x \in (Z_p)^* \Rightarrow x \cdot x^{p-2} = 1 \Rightarrow x^{-1} = x^{p-2}$ in Z_p

another way to compute inverses, but less efficient than Euclid

Dan Boneh

26

Exercise

- Try to compute 2^{10001} mod 11

Dan Boneh

27

Application: generating random primes

Suppose we want to generate a large random prime

say, prime p of length 1024 bits (i.e. $p \approx 2^{1024}$)

Step 1: choose a random integer $p \in [\, 2^{1024} , 2^{1025}-1 \,]$
Step 2: test if $2^{p-1} = 1$ in Z_p
If so, output p and stop. If not, goto step 1 .

Simple algorithm (not the best). **Pr[p not prime] < 2^{-60}**

Dan Boneh

28

Primality test（素性测试）

- **Inputs**: n: a value to test for primality, n>3; k: a parameter that determines the number of times to test for primality
- **Output**: *composite* if n is composite, otherwise *probably prime*
- Repeat k times:
- Pick a randomly in the range $[2, n - 2]$
- If $a^{n-1} \neq 1 \bmod n$, then return *composite*
- If composite is never returned: return *probably prime*

Dan Boneh

29

The structure of $(Z_p)^*$

Thm (Euler): $(Z_p)^*$ is a **cyclic group (循环群)**, that is

$\exists\, g \in (Z_p)^*$ such that $\{1, g, g^2, g^3, \ldots, g^{p-2}\} = (Z_p)^*$

g is called a **generator** (生成元) of $(Z_p)^*$

Example: p=7. $\{1, 3, 3^2, 3^3, 3^4, 3^5\} = \{1, 3, 2, 6, 4, 5\} = (Z_7)^*$

Not every elem. is a generator: $\{1, 2, 2^2, 2^3, 2^4, 2^5\} = \{1, 2, 4\}$

Dan Boneh

30

Euler's generalization of Fermat (1736)

Def: For an integer N define $\varphi(N) = |(Z_N)^*|$ (Euler's φ func.)

Examples: $\varphi(12) = |\{1,5,7,11\}| = 4$; $\varphi(p) = p-1$

For N=p·q: $\varphi(N) = N-p-q+1 = (p-1)(q-1)$

Thm (Euler): $\forall\, x \in (Z_N)^*$: $x^{\varphi(N)} = 1$ in Z_N

Example: $5^{\varphi(12)} = 5^4 = 625 = 1$ in Z_{12}

Generalization of Fermat. Basis of the RSA cryptosystem

Dan Boneh

31

Exercise：验证欧拉定理

- x = 3, N = 10
- x= 2, N = 11
- x = 4, N = 12

Dan Boneh

32

Exercise

- Try to compute 7^{222} mod 10 .

Exponentiation

Finite cyclic group G (for example $G = \mathbf{Z}_p^*$)

Goal: given g in G and x compute g^x

Example: suppose $x = 53 = (110101)_2 = 32+16+4+1$

Then: $g^{53} = g^{32+16+4+1} = g^{32} \cdot g^{16} \cdot g^4 \cdot g^1$

$$g \to g^2 \to g^4 \to g^8 \to g^{16} \to g^{32} \quad g^{53}$$

The repeated squaring alg.

Input: g in G and x>0 ; **Output**: g^x

write $x = (x_n\, x_{n-1} \ldots x_2\, x_1\, x_0)_2$

```
y ← g  ,  z ← 1
for i = 0 to n do:
if (x[i] == 1):    z ← z·y
y ← y²
output z
```

example: g^{53}

y	z
g^2	g
g^4	g
g^8	g^5
g^{16}	g^5
g^{32}	g^{21}
g^{64}	g^{53}

$53 = (110101)_2$

Exercise

- Use the modular exponentiation algorithm to calculate
- （1） 5^9 mod 42
- （2） *11^{13} mod 53*