T-79.159
Cryptography and Data Security

Lecture 6: Modular Arithmetic
-Prime numbers
-Euclid's algorithm
-Chinese remainder theorem
-Euler's totient function
-Euler's theorem

Kaufman et al: Ch 7
Stallings: Ch 8

1

Prime Numbers

Definition: An integer $p > 1$ is a prime if and only if its only positive integer divisors are 1 and p.

Fact: Any integer $a > 1$ has a unique representation as a product of its prime divisors

$$a = \prod_{i=1}^{t} p_i^{e_i} = p_1^{e_1} p_2^{e_2} \cdots p_t^{e_t}$$

where $p_1 < p_2 < \ldots < p_t$ and each e_i is a positive integer.

Some first primes: 2,3,5,7,11,13,17,… For more primes, see:

www.utm.edu/research/**primes**/

Composite (non-prime) numbers and their factorisations:

$18 = 2\times 3^2$, $27 = 3^3$, $42 = 2\times 3\times 7$, $84773093 = 8887\times 9539$

2

Euclid's Algorithm

Given two positive integers and their representations as products of prime powers, it would be easy to extract from them the maximum set of common prime powers.

For example gcd(18, 42) = gcd(2×3^2, $2\times3\times7$) = 2×3 = 6.

However, factoring integers is not an easy task.

Euclid's algorithm is an efficient algorithm for finding the gcd of two integers. It is based on the following fact:

Let $a > b$. Then gcd(a,b) = gcd(a mod b, b).

Example: gcd(42, 18) = gcd(6, 18) = 6.

Example: gcd(595,408) = gcd(187,408) = gcd(187,34) = gcd(17,34) = 17.

Slowest case: Fibonacci sequence 1, 2, 3, 5, 8,13, 21,…,$F_n = F_{n-1} + F_{n-2}$. For example it takes 5 iterations to compute gcd(21,13); in general it takes n-2 iterations to compute gcd(F_n,F_{n-1})

3

Extended Euclidean Algorithm and computing a modular inverse

Fact: Given two positive integers a and b there are integers u and v such that

$$u\times a + v\times b = \gcd(a,b)$$

In particular, if $\gcd(a,b)$ =1, there is a positive integer u such that

$$u\times a = 1 \pmod{b},$$

and similarly, there is a positive integer v such that

$$v\times b = 1 \pmod{a}.$$

u and v can be computed using the Extended Euclidean Algorithm, which iteratively finds integers r_i , u_i and v_i such that

$$r_{i-2} - q_i\times r_{i-1} = r_i \quad \text{and} \quad u_i\times a + v_i\times b = r_i$$

$$u_i = u_{i-2} - q_i \times u_{i-1} \quad \text{and} \quad v_i = v_{i-2} - q_i \times v_{i-1}$$

The index $i = n$ for which r_n = gcd(a,b) gives $u_n = u$ and $v_n = v$.

4

Extended Euclidean Algorithm: Example

gcd(595,408) = 17 = u×595 + v×408

i	q_i	r_i	u_i	v_i
-2	-	595	1	0
-1	-	408	0	1
0	1	187	1	-1
1	2	34	-2	3
2	5	17	11	-16

5

Extended Euclidean Algorithm: Examples

gcd(595,408) = 17 = 11×595 + (-16)×408
= -397×595 + 579×408

We get 11×595 = 17 (mod 408)
and 579×408 = 17 (mod 595)

If gcd(a,b) = 1, this algorithm gives modular inverses.
Example: 557×797 = 1 (mod 1047) that is
557 = 797^{-1} (mod 1047)

If gcd(a,b) = 1, the integers a and b are said to be coprime.

6

Computing multiplicative inverse: Example

gcd(1047,797) = 1 = u×797 + v×1047

i	q_i	r_i	u_i	v_i
-2	-	1047	0	1
-1	-	797	1	0
0	1	250	-1	1
1	3	47	4	-3
2	5	15	-21	16
3	3	2	67	-51
4	7	1	-490	373

7

Chinese Remainder Theorem (two moduli)

Problem: Assume m_1 and m_2 are coprime. Given x_1 and x_2, how to find $0 \le x < m_1 m_2$ such that

$x = x_1 \bmod m_1$

$x = x_2 \bmod m_2$

Solution: Use the Extended Euclidean Algorithm to find u and v such that $u \times m_1 + v \times m_2 = 1$. Then

$x = x \times u \times m_1 + x \times v \times m_2$

$= (x_2 + r \times m_2) \times u \times m_1 + (x_1 + s \times m_1) \times v \times m_2$.

It follows that

$x = x \bmod (m_1 \times m_2) = (x_2 \times u \times m_1 + x_1 \times v \times m_2) \bmod (m_1 \times m_2)$

8

Chinese Remainder Theorem (general case)

Theorem: Assume $m_1, m_2, \ldots, m_t$ are mutually coprime. Denote $M = m_1 \times m_2 \times \ldots \times m_t$. Given $x_1, x_2, \ldots, x_t$ there exists a unique x, $0 < x < M$, such that

$x = x_1 \bmod m_1$

$x = x_2 \bmod m_2$

…

$x = x_t \bmod m_t$

x can be computed as

$x = (x_1 \times u_1 \times M_1 + x_2 \times u_2 \times M_2 + \ldots + x_t \times u_t \times M_t) \bmod M,$

where $M_i = (m_1 \times m_2 \times \ldots \times m_t) / m_i$ and $u_i = M_i^{-1} \pmod{m_i}$

9

Chinese Remainder Theorem: Example

Assume $m_1 = 7$, $m_2 = 11$, $m_3 = 13$. Then $M = 1001$.

Compute x, $0 \le x \le 1000$ such that

$x = 5 \bmod 7$

$x = 3 \bmod 11$

$x = 10 \bmod 13$

$M_1 = m_2 m_3 = 143$; $M_2 = m_1 m_3 = 91$; $M_3 = m_1 m_2 = 77$

$u_1 = M_1^{-1} \bmod m_1 = 143^{-1} \bmod 7 = 3^{-1} \bmod 7 = 5$; similarly

$u_2 = M_2^{-1} \bmod m_2 = 3^{-1} \bmod 11 = 4$; $u_3 = (-1)^{-1} \bmod 13 = -1$.

Then

$x = (5 \times 5 \times 143 + 3 \times 4 \times 91 + 10 \times (-1) \times 77) \bmod 1001 = 894$

10

Euler’s Totient Function $\phi(n)$

Definition: Let n > 1 be integer. Then

$\phi(n)$ = #{ a | 0 < a < n, gcd(a,n) = 1}, that is, $\phi(n)$ is the number of positive integers less than n which are coprime with n.

For prime p, $\phi(p) = p-1$. We set $\phi(1) = 1$.

For a prime power, we have $\phi(p^e) = p^{e-1}(p-1)$

Given m,n, gcd(m,n) = 1, we have $\phi(m\times n) = \phi(m) \times \phi(n)$.

Now Euler’s totient function can be computed for any integer using its prime factorisation.

Example: $\phi(18) = \phi(2\times 3^2) = \phi(2)\times\phi(3^2)= (2-1)\times(3-1)3^1= 6$, that is, the number of invertible numbers modulo 18 is equal to 6. These numbers are: 1,5,7,11,13,17.

11

Euler’s Theorem

$$Z_n^* = \{a \mid 0 < a < n, \gcd(a,n) = 1\}, \text{and } \# Z_n^* = \phi(n)$$

Euler’s Theorem: For any integers n and a such that $a \neq 0$ and gcd(a,n) = 1 the following holds:

$$a^{\phi(n)} \equiv 1 \pmod n$$

Fermat’s Theorem: For a prime p and any integer a such that $a \neq 0$ and a is not a multiple of p the following holds:

$$a^{p-1} \equiv 1 \pmod p$$

12