

NUMBER THEORY AND CRYPTOGRAPHY

OUTLINE:

- 1) Introduction
- 2) Divisibility
- 3) Prime numbers
- 4) Cryptography

1. INTRODUCTION

- Number theory is the part of mathematics which studies the integers and their properties (divisibility, primality, representation in different bases)
- Number theory has deep applications to computer science and cryptography

2. DIVISIBILITY

Divisibility

- Let a and b be integers with $a \neq 0$. We say that a **divides** b (notation $a \mid b$) if there exists an **integer** c such that $b = ac$. In this case, we also say that a is a **multiple** of b and that b is a **divisor**, or a **factor**, of a .
b/a = c
- $a \mid b$ is equivalent to the fact that b/a is an integer.
- EX: for any integer $a \neq 0$, $a \mid 0$: in fact, choosing $c = 0$ in the definition above, we get $0 = a \cdot 0$, or equivalently $0/a = 0$.

Properties of divisibility

- Let a, b, c be integers with $a \neq 0$.
- If $a \mid b$ and $a \mid c$, then $a \mid (b+c)$
- If $a \mid b$ and $a \mid c$, then, for any integers m, n , $a \mid (mb+nc)$
- If $a \mid b$, then $a \mid (b+c)$ whatever c is
- $a \mid a$ (**almost**-reflexivity of \mid : 0 is excluded)
- If $a \mid b$ and $b \mid a$ (which also requires $b \neq 0$), then $a=b$ (antisymmetry of \mid)
- If $a \mid b$ and $b \mid c$, then $a \mid c$ (transitivity of \mid)

$$b = na$$

$$c = mb$$

$$a \mid nma$$

Integer division

- For any integer n (the **dividend**) and $d > 0$ (the **divisor**), there are two uniquely determined integers q (the **quotient**) and r (the **remainder**), with $0 \leq r < d$, such that $n = dq + r$.
- To be clear in the following about integer division vs. real division, we introduce the following notation for integer division:
 - $n \text{ div } b = q = \text{floor}(n/b)$
 - $n \text{ mod } b = r = n - dq$ (the **modulo** operation)
- EX: $105/17 = 6.17647\dots$ (real division)
hence $105 \text{ div } 17 = \underline{6}$ and $105 \text{ mod } 17 = \underline{105 - 6 \cdot 17} = \underline{3}$
 \Rightarrow divisor $\quad \quad \quad \Rightarrow$ remainder

Congruence

- Let $m > 1$ be an integer. Remember that, for two integers a and b , $a \equiv b \pmod{m}$ means that a and b have the same remainder in the integer division by m , that is, $a \bmod m = b \bmod m$
- The relation $\{(a,b) \in \mathbb{Z} \times \mathbb{Z} \mid a \equiv b \pmod{m}\}$ is an equivalence relation on the integers
- Theorem: $a \equiv b \pmod{m}$ iff $m \mid (a-b)$ iff there is an integer k such that $a = b + km$.
 $a - b = mk$

Beware!

- There are 2 almost identical notations which you should not mix up:
- $a = b \bmod m$ (with '=' and boldface '**mod**') denotes the result of an operation: a is the result of b modulo m, i.e., $a = b - m(b \text{ div } m)$
- $a \equiv b \bmod m$ (with '≡' and normal-face 'mod') denotes a relation: a and b are congruent modulo m
- The relationship between the 2 is: $a \equiv b \bmod m$ iff $a \bmod m = b \bmod m$

Properties of congruence

- Let a, b, c, d be integers and let m be a positive integer.
- Congruence mod m is an equivalence relation on the integers:
 - $a \equiv a \pmod{m}$
 - If $a \equiv b \pmod{m}$ then $b \equiv a \pmod{m}$
 - If $a \equiv b \pmod{m}$ and $b \equiv c \pmod{m}$, then $a \equiv c \pmod{m}$
- Compatibility with operations: If $a \equiv b \pmod{m}$ and $c \equiv d \pmod{m}$, then
 - $a + c \equiv b + d \pmod{m}$ (in particular, $a + c \equiv b + c \pmod{m}$)
 - $ac \equiv bd \pmod{m}$ (in particular, $ac \equiv bc \pmod{m}$)

two sets A, B .

All elements of A
has exactly 1 element
corresponded in set B .

Modular arithmetic

- Let $\mathbf{Z}_m = \{0, 1, \dots, m-1\}$. Notice that this set is in obvious bijection with the set $\mathbf{Z}/m\mathbf{Z} = \{[0], [1], \dots, [m-1]\}$ of the equivalence classes of congruence modulo m .
- On \mathbf{Z}_m we define a sum $+_m$ and a product \cdot_m :
 - $a+_mb = a+b \bmod m$ (addition modulo m)
 - $a\cdot_mb = ab \bmod m$ (multiplication modulo m)

Modular arithmetic

Properties of $+_m$ and \cdot_m :

- Associativity: $(a+_mb)+_mc = a+_m(b+_mc)$; $(a\cdot_mb)\cdot_mc = a\cdot_m(b\cdot_mc)$
- Commutativity: $a+_mb = b+_ma$; $a\cdot_mb = b\cdot_ma$
- Neutral elements: $a+_m0 = 0+_ma = a$; $a\cdot_m1 = 1\cdot_ma = a$
- Additive inverses (opposites): $0+_m0 = 0$, and, if $a \neq 0$, $a+_m(m-a) = 0$
- Distributivity of \cdot_m over $+_m$: $(a+_mb)\cdot_m(c+_md) = (a\cdot_mc) +_m (b\cdot_mc) +_m (a\cdot_md) +_m (b\cdot_md)$
- (multiplicative inverses do not always exist: e.g., in arithmetic modulo 6 there is no $a \in \mathbf{Z}_6 = \{0,1,2,3,4,5\}$ such that $3\cdot_6a = 1$)

Modular arithmetic

- Let a, b be integers and let m be a positive integer.
- Iterated modulo operations:
 - $(a + b) \bmod m = ((a \bmod m) + (b \bmod m)) \bmod m = (a \bmod m) +_m (b \bmod m)$
 - $(a \cdot_m b) \bmod m = ((a \bmod m)(b \bmod m)) \bmod m = (a \bmod m) \cdot_m (b \bmod m)$
 - These equalities tell us that, when we perform modular arithmetic, we can either
 - perform all operations in usual arithmetic and take the result **mod** m at the end (fewer operations, but on bigger numbers), or
 - take integers **mod** m right away and then at any intermediate step (more operations, but on smaller numbers).

Modular arithmetic

- EX: arithmetic modulo 6:
 - $1+_63 = 4 \bmod 6 = 4$; $1\cdot_63 = 3 \bmod 6 = 3$
 - $2+_64 = 6 \bmod 6 = 0$; $2\cdot_64 = 8 \bmod 6 = 2$
 - $3+_64 = 7 \bmod 6 = 1$; $3\cdot_64 = 12 \bmod 6 = 0$
 - $(3+_64-_62)\cdot_6(-1) = (7-_62)\cdot_6(-1) = (5)\cdot_6(-1) = -5 \bmod 6 = 1$
(doing calculations in usual arithmetic and taking the modulo at the end)
 - $(3+_64-_62)\cdot_6(-1) = (1-_62)\cdot_6(5) = (5)\cdot_6(5) = 25 \bmod 6 = 1$
(reducing modulo 6 at each step)

Modular arithmetic and congruence classes

- Let a, b be integers and let m be a positive integer.
- The identification of $\mathbf{Z}_m = \{0, 1, \dots, m-1\}$ with $\mathbf{Z}/m\mathbf{Z} = \{[0], [1], \dots, [m-1]\}$ can be used to add and multiply congruence classes modulo m :
- $[a]_m + [b]_m = (a \bmod m) +_m (b \bmod m) = a+b \bmod m = [a+b]_m$
- $[a]_m \cdot [b]_m = (a \bmod m) \cdot_m (b \bmod m) = a \cdot b \bmod m = [a \cdot b]_m$

3. PRIME NUMBERS

The fundamental theorem of arithmetic

- Every natural number greater than 1 is either prime or it can be factored into the product of several primes. The factorization is unique up to the order of the factors.
- In other words, for every natural number n greater than 1, there exist $k \geq 1$ distinct primes p_1, \dots, p_k and k positive integers a_1, \dots, a_k such that

$$n = p_1^{a_1} \cdot p_2^{a_2} \cdot \dots \cdot p_k^{a_k}$$

Examples

- $60 = 2 \cdot 2 \cdot 3 \cdot 5 = 2^2 \cdot 3 \cdot 5$
- $101 = 101$ (it is prime)
- $4096 = 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 = 2^{12}$
- How about negative numbers? The fundamental theorem of arithmetic can be readily adapted: any $n \leq -2$ is either prime or has a factorization into primes (unique up to the order of prime factors)

$$n = -p_1^{a_1} \cdot p_2^{a_2} \cdot \dots \cdot p_k^{a_k}$$

Euclid's theorem

- There are infinitely many primes.
- Proof seen in Logic3.pdf, slide 15.

Finding primes

- The problem of finding large primes is of both theoretical and practical interest.
- So far, no useful closed formula that always produces primes has been found. That is, there is no (known) simple function (sequence) $f(n)$ such that $f(n)$ is prime for all n .
- Many current cryptographic systems are based on the difficulty of
 - Establishing if a large number is prime
 - Factoring a non-prime large number

Sieve of Eratosthenes

- The Sieve of Eratosthenes is an effective, although rather inefficient, method to find all the primes smaller or equal to a given natural number.
- Preliminary observation: if a natural number n is composite, then it must have at least one prime factor $\leq \sqrt{n}$. To see this, note that if $n = ab$, then at least one between a or b has to be $\leq \sqrt{n}$.

Sieve of Eratosthenes

- To find all the primes $\leq N$:
 - List all natural numbers from 2 to N
 - Remove all the multiples of the first number in the list (2), except itself
 - Move to the next number still in the list (3) and remove all its multiples except itself
 - Move to the next number still in the list (5, because 4 has been removed earlier) and remove all its multiples except itself
 - Keep moving to the next number still in the list and removing all its multiples except itself, stopping when that number is $> \sqrt{N}$
 - The remaining numbers are all and only the primes from 2 to N

Sieve of Eratosthenes

- Concrete ex: if $M = 100$:
- Remove the multiples of 2 except 2
- Remove the multiples of 3 except 3
- Remove the multiples of 5 except 5
- Remove the multiples of 7 except 7
- Since $\sqrt{100} = 10$, and at this point the next available number is 11 (8,9,and 10 have been removed), we are done. The remaining numbers are the primes up to 100:
2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97

Mersenne primes

- A prime numbers of the form $2^k - 1$ is called a **Mersenne prime**.
- If k is not prime, surely $2^k - 1$ is not prime either *k is not prime*
- If k is prime, $2^k - 1$ may or may not be prime:
 - $2^5 - 1 = 31$ is prime (ask Eratosthenes)
 - $2^{11} - 1 = 2047 = 23 \cdot 89$ is not prime
- There is an efficient test to determine whether $2^k - 1$ is prime.
- The Great Internet Mersenne Prime Search (GIMPS) is a distributed collaborative project using freely available software to search for large Mersenne prime numbers.

Conjectures about primes

- Many conjectures about prime numbers have been studied for centuries, but remain unresolved
- Goldbach's conjecture: Every even integer $n > 2$; is the sum of two primes. Verified by computers up to $\sim 10^{18}$
- Landau's conjecture: There are infinitely many primes of the form $n^2 + 1$ ($n \in \mathbf{N}$)
- The Twin Prime Conjecture: there are infinitely many pairs of twin primes. Twin primes are pairs of primes that differ by 2 (e.g. 3 and 5, 5 and 7, 11 and 13, etc.)

gcd file.c Greatest common divisor (gcd)

- Let a and b be integers, not both zero. The **greatest common divisor** of a and b , denoted $\gcd(a,b)$ is the largest integer d such that $d \mid a$ and also $d \mid b$.
- The integers a and b are **relatively prime**, or **coprime**, if $\gcd(a,b) = 1$.
- The integers a_1, a_2, \dots, a_n are **pairwise relatively prime**, or **pairwise coprime**, if for any $i \neq j$ $\gcd(a_i, a_j) = 1$.
- EX: Since $\gcd(12,13) = 1$ and $\gcd(13,14) = 1$, 12 and 13 are coprime, and 13 and 14 are coprime as well. However, 12,13 and 14 are not pairwise coprime because $\gcd(12,14) = 2$.

Greatest common divisor (gcd)

- gcds of small numbers can be computed by brute force.
- gcds can be computed immediately if the prime factorizations of the two numbers are known:

if $m = p_1^{a_1} \cdot p_2^{a_2} \cdot \dots \cdot p_k^{a_k}$ and $n = p_1^{b_1} \cdot p_2^{b_2} \cdot \dots \cdot p_k^{b_k}$ where p_1, \dots, p_k are the primes appearing in the factorization of either n or m (so that some exponents may be 0), then

$$\underline{\gcd(m, n) = p_1^{\min(a_1, b_1)} \cdot p_2^{\min(a_2, b_2)} \cdot \dots \cdot p_k^{\min(a_k, b_k)}}$$

Greatest common divisor (gcd)

- EX: since $120 = 2^3 \cdot 3 \cdot 5 = 2^3 \cdot 3 \cdot 5 \cdot 7^0$ and $700 = 2^2 \cdot 5^2 \cdot 7 = 2^2 \cdot 3^0 \cdot 5^2 \cdot 7$, we have

$$\gcd(120, 700) = 2^{\min(3,2)} \cdot 3^{\min(1,0)} \cdot 5^{\min(1,2)} \cdot 7^{\min(0,1)} = 2^2 \cdot 3^0 \cdot 5^1 \cdot 7^0 = 2^2 \cdot 5 = 20$$

- Note that, to compute the gcd with this method, we have “artificially” included the prime 7 in the factorization of 120 (with exponent 0, since 7 is not a prime divisor of 120). Same for 700 and 3. But, as expected, 3 and 7 do not appear in the final factorization of $\gcd(120, 700)$

Greatest common divisor (gcd)

- Finding the gcd through prime factorization is not an efficient method, because there are no known efficient algorithms for the factorization itself.
- Euclid's algorithm for the gcd (which you coded in Assignment 2) is much more efficient, especially the version which uses **mod**.

Euclidean algorithm

- The gcd of 2 integers $a \geq b$ can be computed efficiently using an algorithm devised by Euclid. It amounts at a sequence of integer divisions:
 - $a = b \cdot q_0 + r_0$
 - $b = r_0 \cdot q_1 + r_1$
 - $r_0 = r_1 \cdot q_2 + r_2$
 - $r_1 = r_2 \cdot q_3 + r_3$
 - ...
 - $r_{k-1} = r_k \cdot q_{k+1} + r_{k+1}$ with $r_{k+1} = 0$

terminating when the last remainder (r_{k+1}) is 0. The last nonzero remainder (r_k) is $\gcd(a,b)$.

Euclidean algorithm

- Why does this work?
- Termination: since at every step the remainders decrease towards 0, they eventually reach 0.
- Divisibility
 - r_k divides r_{k-1} because the final remainder is 0
 - r_k divides r_{k-2} because $r_{k-2} = r_{k-1} \cdot q_k + r_k$
 - Iterating the same argument, r_k divides all previous remainders, as well as a and b . Therefore r_k is a common divisor of a and b .
 - Reversing the direction of the above argument, a divisor of a and b is necessarily a divisor of r_k . Hence r_k is the greatest among the common divisors of a and b .

Euclidean algorithm

- Pseudocode: non-recursive version

Algorithm 1 gcd

Require: $a, b \in \mathbb{N}$, $a > b$.

Ensure: gcd(a, b), the GCD of a and b .

while $b \neq 0$ **do**

$t = b$

$b = a \bmod b$

$a = t$

end while

return a

- Pseudocode: recursive version

Algorithm 2 gcd

Require: $a, b \in \mathbb{N}$, $a > b$.

Ensure: gcd(a, b), the GCD of a and b .

if $b = 0$ **then**

return a

else

return gcd($b, a \bmod b$)

end if

Least common multiple (lcm)

- Let a and b be positive integers. The **least common multiple** of a and b , denoted $\text{lcm}(a,b)$ is the smallest natural m such that $a \mid m$ and also $b \mid m$.
- The least common multiple can also be computed from prime factorizations:

if $m = p_1^{a_1} \cdot p_2^{a_2} \cdot \dots \cdot p_k^{a_k}$ and $n = p_1^{b_1} \cdot p_2^{b_2} \cdot \dots \cdot p_k^{b_k}$ where p_1, \dots, p_k are the primes appearing in the factorization of either n or m (so that some exponents may be 0), then

$$\text{lcm}(m, n) = p_1^{\max(a_1, b_1)} \cdot p_2^{\max(a_2, b_2)} \cdot \dots \cdot p_k^{\max(a_k, b_k)}$$

Least common multiple (lcm)

- EX: since $120 = 2^3 \cdot 3 \cdot 5 = 2^3 \cdot 3 \cdot 5 \cdot 7^0$ and $700 = 2^2 \cdot 5^2 \cdot 7 = 2^2 \cdot 3^0 \cdot 5^2 \cdot 7$, we have

$$\text{lcm}(120, 700) = 2^{\max(3,2)} \cdot 3^{\max(1,0)} \cdot 5^{\max(1,2)} \cdot 7^{\max(0,1)} = 2^3 \cdot 3^1 \cdot 5^2 \cdot 7^1 = 4200$$

- Note that, to compute the gcd with this method, we have “artificially” included the prime 7 in the factorization of 120 (with exponent 0, since 7 is not a prime divisor of 120). Same for 700 and 3. In this case, 3 and 7 do appear in the final factorization of $\text{lcm}(120, 700)$

Cute theorem

- Let a and b be positive integers. Then,

$$\underline{a \cdot b = \gcd(a,b) \cdot \text{lcm}(a,b)}$$

- The proof is based on the factorizations of a and b , the properties of powers, and the observation that $\min(a,b) + \max(a,b) = a+b$

Bézout's Theorem

- If a and b are positive integers, then there exist integers s and t such that

$$\gcd(a,b) = sa + tb.$$

- This equation is called **Bézout's identity**.
- The integers s and t are called **Bézout coefficients** of a and b .
- EX: $\gcd(120,700) = 20 = 6 \cdot 120 + (-1) \cdot 700$

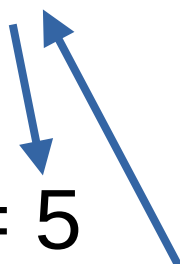
Bézout's Theorem

- Bézout coefficients can be computed working Euclid's algorithm backwards:
- First we apply the Euclid's algorithm to find the gcd
- $\text{Gcd}(135, 145)$
- $145 = 135 \cdot 1 + 10$
- $135 = 10 \cdot 13 + 5$
- $10 = 5 \cdot 2 + 0$

Bézout's Theorem

- Bézout coefficients can be computed working Euclid's algorithm backwards:
- First we apply the Euclid's algorithm to find the gcd

- $\text{Gcd}(135, 145)$
- $145 = 135 \cdot 1 + 10$
- $135 = 10 \cdot 13 + 5$
- $10 = 5 \cdot 2 + \underline{0}$
- $\text{Gcd}(135, 145) = 5$



Two blue arrows originate from the right side of the equations. One arrow points from the '5' in the third equation ($135 = 10 \cdot 13 + 5$) down to the '5' in the final result ($\text{Gcd}(135, 145) = 5$). The other arrow points from the '0' in the fourth equation ($10 = 5 \cdot 2 + \underline{0}$) up to the '5' in the third equation, indicating the step where the remainder becomes zero.

Last
nonzero
remainder

Bézout's Theorem

- Then we use the 1st equation in which the gcd appear (as the remainder)
- $\text{Gcd}(135, 145)$
- $145 = 135 \cdot 1 + 10$
- $135 = 10 \cdot 13 + 5$
- $10 = 5 \cdot 2 + 0$
- $\text{Gcd}(135, 145) = 5$

Bézout's Theorem

- Then we use the 1st equation in which the gcd appear (as the remainder)
- We solve for the remainder (i.e., the gcd):
- $5 = 135 - 10 \cdot 13$
- $\text{Gcd}(135, 145)$
- $145 = 135 \cdot 1 + 10$
- $135 = 10 \cdot 13 + 5$
- $10 = 5 \cdot 2 + 0$
- $\text{Gcd}(135, 145) = 5$

Bézout's Theorem

- Then we solve for the remainder in the previous step of Euclid's algorithm
- $10 = 145 - 135 \cdot 1$
- And we substitute the expression in the previous equation
- $5 = 135 - 10 \cdot 13$
- $5 = 135 - (145 - 135 \cdot 1) \cdot 13$
- $5 = 14 \cdot 135 - 13 \cdot 145$
- $\text{Gcd}(135, 145)$
- $145 = 135 \cdot 1 + 10$
- $135 = 10 \cdot 13 + 5$
- $10 = 5 \cdot 2 + 0$
- $\text{Gcd}(135, 145) = 5$

Bézout's Theorem

- We continue until we reach the 1st step in Euclid's algorithm
- At that point, we have the required Bézout identity (in our case we have already reached this:
 - $5 = 14 \cdot 135 - 13 \cdot 145$
- $\text{Gcd}(135, 145)$
- $145 = 135 \cdot 1 + 10$
- $135 = 10 \cdot 13 + 5$
- $10 = 5 \cdot 2 + 0$
- $\text{Gcd}(135, 145) = 5$

Corollaries of Bézout's Theorem

- If a, b, c are positive integers such that a and b are relatively prime (i.e., $\gcd(a, b) = 1$) and $a \mid bc$, then $a \mid c$.
- Generalization: if a_1, a_2, \dots, a_n are positive integers, p is a prime and $p \mid a_1 \cdot a_2 \cdot \dots \cdot a_n$, then, for some index i , $p \mid a_i$.

Corollaries of Bézout's Theorem

- Dividing both sides of a valid congruence by an integer does not always produce a valid congruence
- However, dividing by an integer which is relatively prime to the modulo does result in a valid congruence:
- Let m be a positive integer and let a, b, c be integers. If $\gcd(m, c) = 1$ and $ac \equiv bc \pmod{m}$, then $a \equiv b \pmod{m}$.

Linear congruences

- A congruence of the form $ax \equiv b \pmod{m}$, where a and b are integers, m is a positive integer, and x is a variable, is called a **linear congruence**.
- The **solutions** of a linear congruence $ax \equiv b \pmod{m}$ are all integers x (not just integers between 0 and $m-1$) that satisfy the congruence.
- An integer z such that $cz \equiv 1 \pmod{m}$ is said to be an **inverse** of c **modulo** m (and c is an inverse of z modulo m).
- EX: 7 is an inverse of 3 mod 20, because $7 \cdot 3 = 21 \equiv 1 \pmod{20}$.
27 is also an inverse of 3 mod 20, because $27 \cdot 3 = 81 \equiv 1 \pmod{20}$.
7 is not an inverse of 3 mod 9, because $7 \cdot 3 = 21 \equiv 3 \pmod{9}$.
QUESTION: can you find an inverse of 3 mod 9?

No

Linear congruences

- Let $ax \equiv b \pmod{m}$ be a linear congruence. If a has an inverse modulo m , say c , then we can solve the congruence multiplying both sides by c :
- $ax \equiv b \pmod{m} \Rightarrow cax \equiv cb \pmod{m} \Rightarrow x \equiv cb \pmod{m}$
- EX: solve $3x \equiv 4 \pmod{20}$
 - 7 is an inverse of 3 modulo 20
 - $3x \equiv 4 \pmod{20} \Rightarrow 7 \cdot 3x \equiv 7 \cdot 4 \pmod{20} \Rightarrow x \equiv 28 \pmod{20} \Rightarrow x \equiv 8 \pmod{20}$

Therefore the solutions are all and only the integers in the congruence class $[8]_{20}$, i.e. ..., -32, -12, 8, 28, 48, ...

Linear congruences

- An inverse of a modulo m exists iff $\gcd(a,m) = 1$
(consequence of Bézout's Theorem)
- EX: find an inverse of 110 modulo 2301
 - 1) Compute $\gcd(110,2301)$ (Euclid)
 - $2301 = 110 \cdot 20 + 101$
 - $110 = 101 \cdot 1 + 9$
 - $101 = 9 \cdot 11 + 2$
 - $9 = 2 \cdot 4 + 1$ $\gcd(110,2301) = 1$, so an inverse exists
 - $2 = 1 \cdot 2 + 0$

$$9 = 2 \cdot 4 + 1$$

$$1 = 9 - 2 \cdot 4$$

$$= (110 - 101) - (101 - 9 \cdot 11) \cdot 4$$

$$= [110 - (2301 - 110 \cdot 20)] - [(2301 - 110 \cdot 20) - (110 - 101) \cdot 11] \cdot 4$$

$$= [110 - (2301 - 110 \cdot 20)] - \{ (2301 - 110 \cdot 20) - [110 - (2301 - 110 \cdot 20)] \cdot 11 \} \cdot 4$$

$$= 110 \cdot 21 - 2301 - [2301 - 110 \cdot 20 - (110 \cdot 21 - 2301) \cdot 11] \cdot 4$$

$$= 110 \cdot 21 - 2301 - [2301 - 110 \cdot 20 - 110 \cdot 231 + 2301 \cdot 11] \cdot 4$$

$$= 110 \cdot 21 - 2301 - 2301 \cdot 48 + 110 \cdot 1004$$

$$= 110 \cdot 1025 - 2301 \cdot 49$$

$$= 2301 \cdot (-49) + 110 \cdot 1025$$

Linear congruences

- An inverse of a modulo m exists iff $\gcd(a,m) = 1$
(consequence of Bézout's Theorem)
 - EX: find an inverse of 110 modulo 2301
- 2) Rearrange the equations solving for the remainders
(we don't need the last one)
- $101 = 2301 - 110 \cdot 20$
 - $9 = 110 - 101 \cdot 1$
 - $2 = 101 - 9 \cdot 11$
 - $1 = 9 - 2 \cdot 4$

Linear congruences

- An inverse of a modulo m exists iff $\gcd(a,m) = 1$ (consequence of Bézout's Theorem)

- EX: find an inverse of 110 modulo 2301

3) Find Bézout coefficients substituting backwards

- $101 = 2301 - 110 \cdot 20$; $9 = 110 - 101 \cdot 1$; $2 = 101 - 9 \cdot 11$; $1 = 9 - 2 \cdot 4$
- $1 = 9 - 2 \cdot 4 = 9 - (101 - 9 \cdot 11) \cdot 4 = 9 \cdot 45 - 101 \cdot 4 = (110 - 101 \cdot 1) \cdot 45 - 101 \cdot 4 = 110 \cdot 45 - 101 \cdot 49 = 110 \cdot 45 - (2301 - 110 \cdot 20) \cdot 49 = 2301 \cdot (-49) + 110 \cdot 1025$

Linear congruences

- An inverse of a modulo m exists iff $\gcd(a, m) = 1$ (consequence of Bézout's Theorem)
 - EX: find an inverse of 110 modulo 2301
- 4) Reduce Bézout's identity modulo 2301
- $1 = 2301 \cdot (-49) + 110 \cdot 1025$ (Bézout's identity)
 - $1 \equiv 110 \cdot 1025 \pmod{2301}$
 - 1025 is the inverse of 110 mod 2301

Linear congruences

- EX: Solve $101x \equiv 4 \pmod{2301}$
 - An inverse of 101 modulo 2301 is 1025
 - So we multiply both sides of the congruence by 1025
 - $1025 \cdot 101x \equiv 1025 \cdot 4 \pmod{2301}$
 - $x \equiv 4100 \pmod{2301}$
 - $x \equiv 1799 \pmod{2301}$

Systems of linear congruences

- Suppose you have to solve a **system of linear congruences**, that is, you need the **common** solutions of 2 or more linear congruences. In the case the modulus are pairwise coprime, an answer is given by the
- Chinese Remainder Theorem: Let m and n be two coprime integers. For every $a, b \in \mathbb{Z}$, there exists $c \in \mathbb{Z}$ such that the system

$$\begin{cases} x \equiv a \pmod{m} \\ x \equiv b \pmod{n} \end{cases}$$

is equivalent to the single congruence $x \equiv c \pmod{mn}$.

Moreover, if a Bézout identity $sm + tn = 1$ is known, then c can be determined as $c = a + (b - a)sm = (a - b)tn + b$

Systems of linear congruences

- EX: find the common solutions of the 2 congruences $x \equiv 1 \pmod{3}$ and $x \equiv 4 \pmod{5}$.
- In the notation of the Chinese Remainder Theorem, we have $m=3$, $n=5$, $a=1$, $b=4$. Moreover, 3 and 5 are coprime, and a Bézout identity is $2 \cdot 3 + (-1) \cdot 5 = 1$, so $s=2$, $t=-1$.
- Then set $c = a + (b-a)sm = 1 + 3 \cdot 2 \cdot 3 = 19$
- The system of the 2 congruences is equivalent to the single congruence $x \equiv 19 \pmod{3 \cdot 5}$, that is, $x \equiv 19 \pmod{15}$, that is, $x \equiv 4 \pmod{15}$, that is, the solutions are all the $x \in \{\dots, -26, -11, 4, 19, 34, \dots\}$

4. CRYPTOGRAPHY

Caesar cipher

- "If he had anything confidential to say, he wrote it in cipher, that is, by so changing the order of the letters of the alphabet, that not a word could be made out. If anyone wishes to decipher these, and get at their meaning, he must substitute the fourth letter of the alphabet, namely D, for A, and so with the others."

— Suetonius, Life of Julius Caesar, 56

- Julius Caesar created secret messages by shifting each letter three letters forward in the alphabet (and sending the last three letters to the first three).
- In symbols, this process can be codified as:
 - The English alphabet has 26 letters. Think of their indices as $\mathbf{Z}_{26} = \{0, 1, \dots, 25\}$.
 - To encrypt a message, use the encryption function $\mathbf{e} : \mathbf{Z}_{26} \rightarrow \mathbf{Z}_{26}$, $\mathbf{e}(x) = x+3 \bmod 26$, that is, replace the letter in position x with the letter in position $x+3 \bmod 26$.
 - To decrypt a message, use the decryption function $\mathbf{d} : \mathbf{Z}_{26} \rightarrow \mathbf{Z}_{26}$, $\mathbf{d}(x) = x-3 \bmod 26$. Note that the decryption function is the inverse of the (bijective) encryption function.

Caesar cipher

- EX: Encrypt the message “Socrates is mortal” with Caesar cipher

- Represent the characters of the message with their index mod 26:

| | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |

“18 14 2 17 0 19 4 18 8 18 12 14 17 19 0 11”

- Apply the encryption function $e(x) = x+3 \bmod 26$ to the numbers in the list

“21 17 5 20 3 22 7 21 11 21 15 17 20 22 3 14”

- Translate the numbers back to letters to get the encrypted message:
“VTFUDWHV LV PRUWDO”

Caesar cipher

- EX: Decrypt the message “LDFWD DOHD HVW” which has been encoded with Caesar cipher

- Represent the characters of the message with their index mod 26:

| | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |

“11 3 5 22 3 3 14 7 3 7 21 22”

- Apply the decryption function $d(x) = x - 3 \bmod 26$ to the numbers in the list

“8 0 2 19 0 0 11 4 0 4 18 19”

- Translate the numbers back to letters to get the encrypted message:

“iacta alea est”

Affine ciphers

- Caesar cipher is a special case of **affine ciphers** which use encryption functions of the form

$$\mathbf{e} : \mathbf{Z}_{26} \rightarrow \mathbf{Z}_{26}, \mathbf{e}(x) = ax+b \bmod 26$$

where a, b are arbitrary integers, with the only condition that the resulting \mathbf{e} be bijective.

- The pair (a, b) is called the **key** of the affine cipher.
- Exercise for you: show that $\mathbf{e}(x) = ax+b \bmod 26$ is bijective iff $\gcd(a, 26) = 1$, and in that case find an inverse function (decryption function).

Affine ciphers

- EX: $\mathbf{e} : \mathbf{Z}_{26} \rightarrow \mathbf{Z}_{26}$, $\mathbf{e}(x) = 3x+5 \bmod 26$ is a valid (i.e. bijective) encryption function for an affine cipher. Encode the message “hello world” using this affine cipher
 - Represent the characters of the message with their index mod 26:

| | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |

“7 4 11 11 14 22 14 17 11 3”
 - Apply the encryption function $\mathbf{e}(x) = 3x+5 \bmod 26$ to the numbers in the list
“0 17 12 12 21 19 21 4 12 14”
 - Translate the numbers back to letters to get the encrypted message:
“ARMMV TVEMO”

Public vs private key

- Affine ciphers are **private key cryptosystems**.
- In a private key cryptosystem, knowing the encryption key allows anyone (friend or foe) to quickly determine the decryption key.
Therefore
 - 1) all parties who wish to communicate have to share the key, and
 - 2) the key has to be kept private.
- In **public key cryptosystems** knowing how to encrypt a message does not help in decrypting the message.
- Therefore, everyone can have a **public encryption key**. The only key that needs to be kept **private** is the **decryption key**.

The RSA cryptosystem

- This is a public key cryptosystem publicly described by Ron Rivest, Adi Shamir and Leonard Adleman in 1977 (an equivalent system had been secretly developed by Clifford Cocks for the British intelligence services).
- The encryption key (public) is a pair of integers (n, e) , where $n = p \cdot q$ (the **modulus**) is the product of two large primes, and e (the **exponent**) is coprime with the product $(p-1) \cdot (q-1)$.

The RSA cryptosystem

- Choose 2 primes p, q and keep them **secret**.
- Compute $n = pq$ and **release** it as part of the public key.
- Compute $k = \text{lcm}(p-1, q-1)$ (e.g., Euclid algorithm + cute theorem) and keep it **secret**.
- Choose an integer e such that $1 < e < k$ and $\gcd(e, k) = 1$ and **release** it as part of the public key.
- Compute $d \equiv e^{-1} \pmod{k}$ (e.g., Euclid + Bézout: d is a Bézout coefficient for e in the Bézout identity $de + tk = 1 = \gcd(e, k)$) and keep it **secret**.

The RSA cryptosystem

- The encryption function is $\mathbf{e}(x) = x^e \bmod n$ (anybody can reasonably quickly encrypt a message with this function because e and n are known)
- The decryption function is $\mathbf{d}(y) = y^d \bmod n$ (only people who know d can use the decryption function; it is unreasonably long to find d without knowing k (or equivalently p and q))
- To encrypt the message x , we compute $y = \mathbf{e}(x) = x^e \bmod n$ (y is the encrypted message)
- To decrypt y , whoever knows d computes $x = \mathbf{d}(y) = y^d \bmod n = (x^e)^d \bmod n$

The RSA cryptosystem

- The decryption function works because for any x (plain message) $(x^e)^d = x^{ed} \equiv x \pmod{n}$ (consequence of Fermat's Little Theorem applied to p and q , and of the Chinese Remainder Theorem)
- RSA is a solid cryptosystem since the only known method of finding d is based on a factorization of n into primes, and there is currently no known efficient algorithm to factor large numbers into primes.