Manchester Metropolitan University

Introduction to Number Theory

Killian O’Brien

6G6Z0024 Applied Cryptography

Lecture Week 01 – Mon 29 September 2025

Introduction to the unit

- Teaching team: Dr Killian O’Brien and Dr Safiullah Khan . See Moodle for contact details.
- 6G6Z0024 Applied Cryptography (15 credits)
- Assessment is 100% coursework. A portfolio of exercises.
- Timetable
- Let’s look at the Moodle page for the unit.

Introduction to Number Theory

We deal with the positive and negative *counting* numbers, more properly named the *integers*, and denoted by the symbol $\mathbb{Z}$, (coming from the German *Zahl*, for number)

- $\mathbb{Z} = \{\ldots, -3, -2, -1, 0, 1, 2, 3, \ldots\}$
- $\mathbb{Z}$ is an *infinite* set.
- $\mathbb{Z}$ obviously carries the operations of addition, $+$, and multiplication, $\cdot$, that you've known from primary school.

Modern cryptography relies heavily on techniques and facts from *number theory*, which is the mathematical study of the integers and their properties under $+$ and $\cdot$.

Topics we need to know

- The **divisibility relation** on $\mathbb{Z}$.
- **Greatest Common Divisors** (gcd) and the **Euclidean Algorithm**.
- The **congruence relation** and **modular arithmetic**.
- **Prime** numbers and
  - The **Fundamental Theorem of Arithmetic** and **prime factorizations**
  - **Fermat's Little Theorem**
  - **Euler's totient** function
  - **Euler's theorem**
  - **Primality** testing, the **Miller-Rabin** test
- **The Chinese Remainder Theorem**
- **Discrete logarithms**

All these covered in Stallings, Chapter 2: Introduction to Number Theory.

The divisibility relation

- Recall, a *relation* in computer science / mathematics is a formula $A(x_1, \dots, x_n)$, so that when values are supplied for the variables $x_1, \dots, x_n$, results in a *statement* $A(x_1, \dots, x_n)$, i.e. something which is true or false.
- For a pair of integers a, b, with $b \neq 0$, we say b *divides* a, and write $b \mid a$ if there exists an integer c such that

  $$a = b \cdot c,$$

  and if no such integer c exists then we say b does *not divide* a, and can write $b \nmid a$.
- So $b \mid a$ is a binary relation on a, b, i.e. a statement that is true or false, depending on the values of a, b.
- If $b \mid a$ then we say b is a *factor* or *divisor* of a.

Examples

- $3 \mid 15, 5 \mid 15, 1 \mid 15, 15 \mid 15.$
- $3 \nmid 10, 17 \nmid 20.$

Properties of divisibility

The divisibility relation enjoys the following properties, which can all be demonstrated (and proved) using its definition and basic properties of the integers.

- If $a|1$ then $a = \pm 1$, i.e. $a = -1$ or $a = +1$.
- If $a|b$ and $b|a$ then $a = \pm b$.
- For all non-zero integers b, we have $b|0$, i.e. *everything divides 0*.
- If $a|b$ and $b|c$ then $a|c$, i.e. the divisibility relation is *transitive*, it travels through intermediaries.
- If $x|y$ and $x|z$ then for all pairs of integer coefficients α, β, we have

  $$x|(\alpha \cdot y + \beta \cdot z),$$

  i.e. x divides all *linear combinations* of y and z.

To familiarise yourself with these, work through some examples of the transitivity of divisibility and the divisibility of linear combinations.

The integer division algorithm

Do you remember this kind of thing from primary school?

- 20 divided by 3, goes in 6 times, with remainder 2.
- $20 = 6 \cdot 3 + 2$

The *integer division algorithm* is simply a formalization of this. It is:

- Given any positive integer n and any non-negative integer a, we can divide a by n to get an integer quotient q and remainder r that satisfy
- $a = qn + r$, and $0 \leq r < n$, and $q = \lfloor a/n \rfloor$
- $\lfloor x \rfloor$ is defined as the largest integer less than x, the so-called *floor* function.

Greatest Common Divisors (gcd)

We write $\gcd(a, b)$ for the *greatest common divisor of a and b*. So $\gcd$ is defined by

- $\gcd(a, b) = d$, where d is the largest integer that divides both a and b.
- For neatness, we also define $\gcd(0, 0) = 0$.

For example

- $\gcd(60, 24) = 12$, $\gcd(100, 75) = 25$, $\gcd(15, 32) = 1$.
- Note that, by its definition, $\gcd$ will always be non-negative, i.e. $\gcd(-60, 24) = 12$.

For small arguments a, b, we can calculate $\gcd(a, b)$ *in our heads*, so to speak.

- $\gcd(25, 3) =?, \gcd(99, 27) =?, \ldots$.
- But what about $\gcd(12349878973245, 324765)$?

The Euclidean Algorithm

In fact there is a classic algorithm that can quickly determine $\gcd$, and establishes the following, non-obvious fact,

- $\gcd(a, b)$ is the smallest positive integer d that can be written in the form

$$d = x \cdot a + y \cdot b,$$

for integer coefficients x, y.

The Euclidean algortihm was known to ancient mathematicians and has several important uses and generalisations in mathematics and cryptography.

The Euclidean Algorithm

A detailed treatment is given in Stallings. The algorithm depends on the following property of $\gcd$.

- If $a = qn + r$ then $\gcd(a, n) = \gcd(n, r)$.

This is true because

- if d is a common divisor of a and n, then since $r = a - qn$, i.e. r is a linear combination of a and n, then d divides r also. And so d is a common divisor of n and r.
- Similarly we can show that if e is a common divisor of n and r, then e divides a also. And so e will be a common divisor of a and n.
- So the pairs (a, n) and (n, r) have the exact same set of common divisors.
- Therefore,

$$\gcd(a, n) = \gcd(n, r).$$

The Euclidean Algorithm

The algorithm works by repeatedly applying the property from the last slide, to a sequence of integer divisions, until the $\gcd$ is clear. Best seen with a worked example

- What is $\gcd(710, 310)$?
- $710 = 2 \cdot 310 + 90$ so $\gcd(710, 310) = \gcd(310, 90)$,
- $310 = 3 \cdot 90 + 40$ so $\gcd(310, 90) = \gcd(90, 40)$,
- $90 = 2 \cdot 40 + 10$ so $\gcd(90, 40) = \gcd(40, 10)$,
- $40 = 4 \cdot 10 + 0$ so $\gcd(40, 10) = \gcd(10, 0) = 10$.

Note that

- The algorithm will terminate, since the remainders are a strictly decreasing sequence of non-negative integers.
- By definition of divisibility, $\gcd(x, 0) = x$, for all integers x.
- The $\gcd$ equations associated to the integer divisions all link together.
- So we can conclude that

$$\gcd(710, 310) = 10.$$

See Stallings for the full detail, a flowchart specification of the algorithm, and more examples.

The mod operator and the congruence relation

For an integer a and a positive integer n we say that a *modulo* n is the remainder r in the integer division of a by n.

- $a = qn + r, \quad 0 \leq r < n$
- We write $(a \mod n) = r$.
- n is called the *modulus* in this expression.

For example

- $(11 \mod 7) = 4$ and $(-11 \mod 4) = 1$.

There is an associated binary relation here.

We say that two integers a and b are *congruent modulo* n, written as

$$a \equiv b \pmod{n},$$

if

- $(a \mod n) = (b \mod n)$
- That is, if a and b *leave the same remainder*, after division by n.

The mod operator and the congruence relation

Examples

- $23 \equiv 8 \pmod{5}$
- $-11 \equiv 5 \pmod{8}$
- $81 \equiv 0 \pmod{27}$

The congruence relation has the following properties

- $a \equiv b \pmod{n}$ if and only if $n|(a-b)$
- $a \equiv a \pmod{n}$,   called *reflexivity*
- $a \equiv b \pmod{n}$ implies that $b \equiv a \pmod{n}$,   called *symmetry*
- If $a \equiv b \pmod{n}$ and $b \equiv c \pmod{n}$, then $a \equiv c \pmod{n}$,   called *transitivity*
- These last three properties mean congruence modulo n is an *equivalence relation* on $\mathbb{Z}$.

Modular arithmetic

- The mod operator $(a \mod n)$ maps all integers a into the set

$$\mathbb{Z}_n = \{0, 1, 2, 3, \ldots, n-1\}.$$

- This is the set of *residues*, or *remainders*, modulo n.
- The familiar operations of $+$ and $\cdot$ on $\mathbb{Z}$ extend to $\mathbb{Z}_n$ in a natural way.

$$(a \mod n) + (b \mod n) := ((a+b) \mod n)$$

$$(a \mod n) \cdot (b \mod n) := ((a \cdot b) \mod n)$$

This means that $\mathbb{Z}_n$, with the operations of $+$ and $\cdot$ will form a *closed system* with respect to these operations, i.e. for any pair x, y from $\mathbb{Z}_n$, $x + y$ and $x \cdot y$ will again be elements of $\mathbb{Z}_n$.

See Stallings for worked examples of $\mathbb{Z}_8$ under $+$ and $\cdot$.

Modular arithmetic

- So given x from $\mathbb{Z}_n$, x will have an *additive inverse*, $n - x$, which satisfies

$$x + (n - x) \equiv 0 \pmod{n}.$$

- Given x from $\mathbb{Z}_n$, if there exists a y in $\mathbb{Z}_n$ which satisfies

$$x \cdot y \equiv 1 \pmod{n},$$

  then we say y is the *multiplicative inverse of* x *modulo* n, and vice versa. We can write $y \equiv x^{-1} \pmod{n}$.
- But multiplicative inverses do not necessarily exist for every element of $\mathbb{Z}_n$.

Modular arithmetic

This is connected to the issue of cancellation in $\mathbb{Z}_n$.

- If $(a+b) \equiv (a+c) \pmod{n}$ then $b \equiv c \pmod{n}$.
- If $(a \cdot b) \equiv (a \cdot c) \pmod{n}$ then it's not necessarily true that $b \equiv c \pmod{n}$.
- However, if $a^{-1} \pmod{n}$ exists then we can cancel from products as

$$a^{-1}(a \cdot b) \equiv a^{-1}(a \cdot c) \pmod{n}$$

and so

$$(a^{-1}a) \cdot b \equiv (a^{-1}a) \cdot c \pmod{n}$$

and so

$$b \equiv c \pmod{n}.$$

Extended Euclidean algorithm and multiplicative inverses

Using linear combinations and the Euclidean algorithm we can show that

- for a in $\mathbb{Z}_n$, a multiplicative inverse of a modulo n will exists if and only if $\gcd(a, n) = 1$.
- **Terminology**: If $\gcd(a, b) = 1$ then a, b are said to be *relatively prime*, or *coprime*.

Extended Euclidean algorithm

- If $\gcd(a, b) = d$ then the *Extended Euclidean Algorithm* is an algorithm that produces integer coefficients x, y such that

$$xa + yb = d.$$

- It works by using the integer division with remainder equations from the Euclidean algorithm applied to $\gcd(a, b)$ and a method of *back substitution* to generate the x, y.
- We will demonstrate it by finding x, y so that $x \cdot 710 + y \cdot 310 = 10$.

Connection with multiplicative inverses

- When $\gcd(a, n) = 1$ and we find coefficients x, y such that

$$xa + yn = 1,$$

  then $x \equiv a^{-1} \pmod{n}$.