Manchester
Metropolitan
University

Introduction to Number Theory

Killian O’Brien

6G6Z0024 Applied Cryptography 2023/24

Lecture Week 02 – Mon 09 October 2023

Introduction

- My name is Dr Killian O’Brien
- Contacts: k.m.obrien@mmu.ac.uk, Teams chat, Office JDE 114a (first floor of John Dalton East, Chester St end)
- Office hours are Thurs 2-3pm & Thur 4-5pm - before/after your Thurs lecture. I intend to be in the *Learning Studio*, come find me there.
- 6G6Z0024 Applied Cryptography (15 credits)
- The Moodle page for the unit.
- The slides from the first lecture are repeated after this one for reference. New slides for lecture 02 begin here
- Stallings, Chapter 2: Introduction to Number Theory

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 $A(x_1, \ldots, x_n)$, so that when values are supplied for the variables $x_1, \ldots, x_n$, results in a *statement* $A(x_1, \ldots, x_n)$, 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 $b|a$ 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 $b \not| a$.
- So $b|a$ is a binary relation on a, b, i.e. a statement that is true or false, depending on the values of a, b.
- If $b|a$ then we say b is a *factor* or *divisor* of a.

Examples

- $3|15$, $5|15$, $1|15$, $15|15$.
- $3 \not| 10$, $17 \not| 20$.

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 postitive 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 alrgest 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 postitive 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 severl 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.

- $a = qn + r, 0 \leq r < n$
- 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 $(a + b) \equiv (a + c) \pmod{n} thenb \equiv c \pmod{n}$.
- If $(a \cdot b) \equiv (a \cdot c) \pmod{n}$ then it’s not neccessarily 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 $\gcd(x, y) = 1$ then x, y are said to be *relatively prime*, or *coprime*.

See Stallings chapter 2 for details.

Prime numbers

Of central importance in cryptography, and of great interest to mathematicians, are the *prime integers*.

- **Definition - prime**

  And integer $p > 1$ is *prime* if its only positive divisors are 1 and p.
- **Definition - composite**

  A positive integer that is not prime is called a *composite* integer.
- The sequence of primes begins

$$p = 2, 3, 5, 7, 11, 13, 17, 19, 23, \ldots$$

- In fact, there are **infinitely many** primes. Known to Euclid, circa 2,300 years ago. See this Numberphile video for an accessible discussion of the proof of this, and its history.
- The largest prime known to humans is currently

$$2^{82,589,933} - 1,$$

  an integer with approximately 24 million digits. Discovered in 2018, thanks to the GIMPS project.

Primes make integers

Primes are central to number theory thanks to

> **Theorem (Fundamental Theorem of Arithmetic)** Every postive integer $n > 1$ can be written, uniquely, as a product of prime numbers,
>
> $$n = p_1^{a_1} \cdot p_2^{a_2} \cdot \cdots \cdot p_r^{a_r},$$
>
> where the p_i are primes and each a_i is a positive integer exponent.

- The expression in the theorem is known as the *prime factorization of* n and can be written compactly as

$$n = \prod_{i=1}^{r} p_i^{a_i}.$$

- The existence of prime factorizations follows immediately from the definition of a prime, i.e. *keep factoring* n *until you can factor no more.*
- The uniqueness part requires some careful mathematical argument.

Examples

- $91 = 7 \cdot 13$
- $3600 = 2^4 \cdot 3^2 \cdot 5^2$
- $1101 = 7 \cdot 11^2 \cdot 13$

Fermat’s Little Theorem

We need to understand the behaviour of *multiplication* and *exponentiation* on $\mathbb{Z}_n$. **Euler’s Theorem** is a result that tells us a lot about how it behaves. A simpler first case to look at it called **Fermat’s Little Theorem**.

> **Theorem (Fermat's Little Theorem)** If p is a prime and a is a postive integer not divisible by p, (i.e. $a \not\equiv 0 \pmod p$) then
>
> $$a^{p-1} \equiv 1 \pmod p.$$

- A proof for this is given in Stallings.

Fermat’s Little Theorem

Example

With $a = 7$ and 19 we see

$$7^2 \equiv 49 \equiv 11 \pmod{19} \quad 7^4 \equiv (7^2)^2 \equiv 11^2 \equiv 121 \equiv 7 \pmod{19}$$

$$7^8 \equiv 7^2 \equiv 49 \equiv 11 \pmod{19} \quad 7^{16} \equiv 11^2 \equiv 121 \equiv 7 \pmod{19}$$

So now

$$a^{p-1} \equiv a^{16} \equiv a^{16+2} \equiv a^{16} \cdot a^2 \equiv 7^{16} \cdot 7^2 \equiv 7 \cdot 11 \equiv 77 \equiv 1 \pmod{19}.$$

- These calculations show an example of dealing with large exponents, (i.e. 16), by the method of **repeated squares**. More later.

Euler’s totient function

- Recall, the integer a has a multiplicative inverse modulo n if and only if $\gcd(a, n) = 1$, i.e. a and n are coprime (to each other).
- **Definition - Euler's totient function**

  Euler’s totient function $\phi : \mathbb{Z}^+ \to \mathbb{Z}^+$ is defined as: $\phi(n)$ is the number of positive integers a, less than n, (i.e. $1 \leq a < n$) such that $\gcd(a, n) = 1$.

Example

- $\phi(35) = 24$ as the integers coprime to 35 are

$$1, 2, 3, 4, 6, 8, 9, 11, 12, 13, 16, 17, 18,$$

$$19, 22, 23, 24, 26, 27, 29, 31, 32, 33, 34,$$

  and there are 24 integers on this list.
- Notice that $35 = 5 \cdot 7$ and this list omits all multiples of 5 and 7.
- This points to a more systematic way of evaluating $\phi(n)$.

Euler’s totient function

Some evaluation formulae for ϕ are

- For a prime p,

$$\phi(p) = p - 1.$$

- For a power of a prime, p^a, we have

$$\phi(p^a) = p^{a-1}(p-1).$$

- ϕ is *multiplicative*, i.e.

$$\text{if } \gcd(a,b) = 1 \text{ then } \phi(a \cdot b) = \phi(a) \cdot \phi(b).$$

Euler’s totient function

- Putting all these together, means that for an integer n with a prime factorization

  $$n = \prod_{i=1}^{r} p_i^{a_i} = p_1^{a_1} \cdot p_2^{a_2} \cdot \cdots \cdot p_r^{a_r},$$

  then

  $$\phi(n) = \prod_{i=1}^{r} p_i^{a_i - 1}(p_i - 1) = p_1^{a_1 - 1} \cdot (p_1 - 1) \cdot p_2^{a_2 - 1} \cdot (p_2 - 1) \cdot \cdots \cdot p_r^{a_r - 1}(p_r - 1).$$

Example

$$\phi(35) = \phi(5 \cdot 7) = 5^0 \cdot 4 \cdot 7^0 \cdot 6 = 4 \cdot 6 = 24.$$

Euler’s theorem

Finally, we can now state Euler’s theorem, which is a generalization of Fermat’s Little Theorem

Theorem (Euler's Theorem) If n is a postive integer modulus and a and n are coprime then

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

- A proof for this is given in Stallings.
- Theorem allows one to simplify powers of a modulo n, where the exponent is very large.
- Suppose that $b \equiv r \pmod{\phi(n)}$. Think of b being very large and r being relatively small.
- So $b = q \cdot \phi(n) + r$.
- Then

$$a^b = a^{q \cdot \phi(n)+r} = a^{q \cdot \phi(n)} \cdot a^r = \left(a^{\phi(n)}\right)^q \cdot a^r \equiv 1^q \cdot a^r \equiv a^r \pmod{n}.$$

Example

- Suppose $\gcd(a, 35) = 1$ and we want $a^{23458973249848} \pmod{35}$. Remember $\phi(35) = 24$.
- $23458973249848 \equiv 16 \pmod{24}$.
- So $a^{23458973249848} \equiv a^{16} \pmod{35}$.

The Chinese Remainder Theorem

- The CRT is another useful result for working with modular arithmetic.
- Suppose M is an integer factorized into *pairwise coprime* factors

$$M = \prod_{i=1}^{k} m_i = m_1 \cdot m_2 \cdot \ldots m_k,$$

  i.e. $\gcd(m_i, m_j) = 1$ for every pair of distint indices $1 \leq i, j, \leq k, i \neq j$.
- Such a factorization might be given by the different powers of primes in the prime factorization of M, i.e.

$$M = \prod_{i=1}^{k} p_i^{a_i} = (p_1^{a_1}) \cdot (p_2^{a_2}) \cdot \cdots \cdot (p_k^{a_k}).$$

- The CRT describes a *one to one* mapping from the integers $\mathbb{Z}_M$ to the *Cartesian product*,

$$\mathbb{Z}_{m_1} \times \mathbb{Z}_{m_2} \times \cdots \times \mathbb{Z}_{m_k}.$$

- An element of $\mathbb{Z}_{m_1} \times \mathbb{Z}_{m_2} \times \cdots \times \mathbb{Z}_{m_k}$ is a k-tuple

$$(a_1, a_2, \ldots, a_k),$$

  where each a_i is a residues/remainder modulo m_i.

The Chinese Remainder Theorem

- $M = \prod_{i=1}^{k} m_i = m_1 \cdot m_2 \cdot \ldots m_k,$
- $\gcd(m_i, m_j) = 1$ for every pair of distint indices $1 \leq i, j, \leq k, i \neq j$.
- The mapping $\mathbb{Z}_M \to \mathbb{Z}_{m_1} \times \mathbb{Z}_{m_2} \times \cdots \times \mathbb{Z}_{m_k}$ is defined by

$$x \mapsto \Big((x \pmod{m_1}), (x \pmod{m_2}), \ldots, (x \pmod{m_k})\Big)$$

  i.e. reduct x modulo m_i for the i^{th}-component of the k-tuple.
- The mapping in the other direction $\mathbb{Z}_{m_1} \times \mathbb{Z}_{m_2} \times \cdots \times \mathbb{Z}_{m_k} \to \mathbb{Z}_M$ is a little more involved
- For each $1 \leq i \leq k$, define $M_i = M/m_i$, and let M_i^{-1} be the multiplicative inverse of M_i modulo m_i.
- Then, the k-tuple $(a_1, a_2, \ldots, a_k)$ will be mapped to the element a of $\mathbb{Z}_M$ defined by

$$a = \sum_{i=1}^{k} a_i \cdot M_i \cdot M_i^{-1} \pmod{M}.$$

- These two maps described above are *inverses* of one another.
- Arithmetic operations on elements of $\mathbb{Z}_M$ can be achived by corresepoding operations on elements of $\mathbb{Z}_{m_1} \times \mathbb{Z}_{m_2} \times \cdots \times \mathbb{Z}_{m_k}$.
- We will work with examples of this in the today's lab.

Suggested reading

- During the week, read up on the following sections of Chapter 2 from Stallings.
    - Testing for Primality
    - Discrete Logarithms
- Discrete logarithms will be needed for public key encryption. I will cover it then also.
- Next week, our first encryption system, the *Data Encryption Standard* (DES).

// reveal.js plugins