Topic 3: Number Theory

Seppe Staelens

June 25, 2024

Introduction

In this topic, we revisit some very basic concepts which you have known of since primary school: divisibility and prime numbers. These concepts form the foundations of number theory, which is the study of the properties of numbers. We will however quickly discover new concepts, such as congruences and the Euler totient function. Most of the theory is assembled in one long build-up to its application in cryptography, in the form of the RSA algorithm.

1 Lecture 1: Number theory and modular arithmetic

1.1 Divisibility and congruences

You should be familiar with the concept of divisibility from primary school. A number a is divisible by a number b if there exists an integer c such that $a = bc$. You should also be familiar with the concept of the remainder of a division:

Lemma 1. *Let $a \in \mathbb{Z}, b \in \mathbb{Z}_0$. Then there exist unique $q, r \in \mathbb{Z}$ such that*

$$a = bq + r \quad \textit{and} \quad 0 \leq r < |b| \,. \tag{1}$$

*The number q is called the **quotient** of the division of a by b, and r is called the **remainder**.*

The remainder r is often denoted by $a \mod b$. We say that x is **congruent** to y modulo n if $x \mod n = y \mod n$, i.e. if they have the same remainder after division by n. This is often denoted as

$$x \equiv y \mod n \,.$$

Proposition 1. *Let $x, y \in \mathbb{Z}$ and $n \in \mathbb{N}_0$. Then the following equivalence holds:*

$$x \equiv y \mod n \quad \Leftrightarrow \exists k \in \mathbb{Z} : x = y + kn \,. \tag{2}$$

Proof. The proof is straightforward and left as an exercise. □

The following proposition is useful for calculations with congruences.

Proposition 2. *Let $x, x', y, y' \in \mathbb{Z}$ and $n \in \mathbb{N}_0$. Then the following equalities and equivalences hold:*

1. $(x + y) \mod n = (x \mod n + y \mod n) \mod n$.
2. $(x \cdot y) \mod n = (x \mod n \cdot y \mod n) \mod n$.
3. $x \equiv x' \mod n$ *and* $y \equiv y' \mod n \Rightarrow x + y \equiv x' + y' \mod n$.

4. $x \equiv x' \mod n$ *and* $y \equiv y' \mod n \Rightarrow xy \equiv x'y' \mod n$.

Proof. The proof is straightforward and left as an exercise. □

These rules are very useful when doing concrete calculations, as we can always reduce numbers to smaller values before doing the actual calculations.

Exercise 1. *Calculate* $12345^6 \mod 7$.

Exercise 2. *In primary school you may have learned a trick to check whether a number is divisble by 9, namely by checking whether the sum of its digits is divisible by 9. Can you explain this using congruences?*

You should also be familiar with the concept of **prime numbers**.

Definition 1. *A number* $p \in \mathbb{N}$ *is called prime if it has exactly two divisors:* 1 *and* p *itself.*

Prime numbers are ubiquitous in number theory, and one could spend an entire summer school just on their properties and mysteries. Quite a number of applications become much more interesting when the numbers involved are prime, as will be the case for cryptography as well.

We mention the following two essential results **SJS: prove?**

Theorem 1. *Every number* $n \in \mathbb{N}$ *can be uniquely written as a product of prime numbers.*

Theorem 2. *There are infinitely many prime numbers.*

Another interesting concept in number theory is that of the **greatest common divisor** of two numbers.

Definition 2. *Let* $a, b \in \mathbb{Z}$. *The greatest common divisor of* a *and* b, *denoted* $\gcd(a, b)$, *is the largest number that divides both* a *and* b.

Definition 3. *Two numbers* $a, b \in \mathbb{Z}$ *are called* ***coprime*** *if* $\gcd(a, b) = 1$.

These concepts are at the basis of following famous theorem. **SJS: necessary?**

Theorem 3. *(**Chinese Remainder Theorem**) Let* $n_1, n_2, \ldots, n_r \in \mathbb{N}_0$ *be coprime. Then,* $\forall a_1, a_2, \ldots, a_r \in \mathbb{Z}$, *the system of congruences*

$$\begin{cases} x & \equiv a_1 \mod n_1\,, \\ x & \equiv a_2 \mod n_2\,, \\ & \vdots \\ x & \equiv a_r \mod n_r\,, \end{cases}$$

has a solution $x \in \mathbb{Z}$. *Furthermore, all solutions are congruent modulo* $n_1 n_2 \cdots n_r$, *i.e. if a solution* $x_0 \in \mathbb{Z}$ *is known, all other solutions are of the form*

$$x_0 + k n_1 n_2 \cdots n_r\,,$$

for $k \in \mathbb{Z}$.

Proof. TBD □

SJS: we need this one for one of the cryptography questions.

Corollary 1. *Let* $m, n \in \mathbb{N}_0$ *be coprime. Then for all* $x, x' \in \mathbb{Z}$ *the following holds:*

$$x \equiv x' \mod mn \quad \Leftrightarrow \quad \begin{cases} x & \equiv x' \mod m\,, \\ x & \equiv x' \mod n\,. \end{cases} \tag{3}$$

SJS: We need this one for one of the cryptography questions.

Theorem 4. *(**Bézout-Bachet**) Let* $a, b \in \mathbb{Z}$. *Then there exist* $x, y \in \mathbb{Z}$ *such that* $ax + by = \gcd(a, b)$.

Proof. TBD □

1.2 Modular arithmetic

At the start of the topic on algebra, we have discussed the concept of a group. Again, we will impose more structure on a group to obtain a new algebraic structure, called a ring.

Definition 4. *A **ring** is a set R with two binary operations $+$ and $\cdot$ such that:*

1. *$(R, +)$ is an abelian group.*
2. *The operation $\cdot$ is associative, i.e. $\forall x, y, z \in R : x \cdot (y \cdot z) = (x \cdot y) \cdot z$.*
3. *The operation $\cdot$ is distributive over $+$, i.e. $\forall x, y, z \in R : x \cdot (y+z) = x \cdot y + x \cdot z$ and $(x+y) \cdot z = x \cdot z + y \cdot z$.*

The ring is sometimes denoted as $(R, +, \cdot)$.

Definition 5. *The Euler totient function $\phi : \mathbb{N} \to \mathbb{N}$ is defined as the number of positive integers less than n that are coprime to n.*

We state the following theorem without proof, as it would require a more in-depth discussion of rings.

Theorem 5. *Take $m, n \in \mathbb{N}_0$ with $\gcd(m, n) = 1$. Then $\phi(mn) = \phi(m)\phi(n)$.*

This is a very useful theorem, as it allows us to calculate $\phi(n)$ for any n by factorizing n into its prime factors. Indeed, suppose

$$n = p_1^{e_1} \cdots p_r^{e_r} \,,$$

then

$$\phi(n) = \phi(p_1^{e_1}) \cdots \phi(p_r^{e_r}) \,,$$

simply because prime numbers are by definition coprime.

The following lemma gives us the last piece of the puzzle to fully calculate $\phi(n)$.

Lemma 2. *For any prime number p and any $e \in \mathbb{N}_0$, we have*

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

Proof. We count the number of elements in $\{1, 2, \ldots, p^e\}$ that are coprime to p^e. The only numbers that are not coprime to p^e are the multiples of p, i.e.

$$p, 2p, 3p, \ldots, p^{e-1}p \,.$$

There are p^{e-1} such multiples, so there are $p^e - p^{e-1} = (p-1)p^{e-1}$ numbers that are coprime to p^e. □

The following theorem summarizes the above.

Theorem 6. *For any $n \in \mathbb{N}_0$, given its prime factorization $n = p_1^{e_1} \cdots p_r^{e_r}$, we have that*

$$\phi(n) = (p_1 - 1)\, p_1^{e_1 - 1} \cdots (p_r - 1)\, p_r^{e_r - 1} \,. \tag{4}$$

So, the Euler totient function of any natural number can be calculated once its prime factors are known.

We state the following theorem without proof, as it requires more knowledge of rings.

Theorem 7. ***(Euler's congruence)*** *Let $a \in \mathbb{Z}, n \in \mathbb{N}_0$ be coprime. Then*

$$a^{\phi(n)} \equiv 1 \mod n \,. \tag{5}$$

Corollary 2. *Let $a \in \mathbb{Z}, n \in \mathbb{N}_0$ be coprime. Then $\forall e \in \mathbb{N}_0$*

$$a^e \equiv a^{e \mod \phi(n)} \mod n \,. \tag{6}$$

Proof. The proof follows from Thm. 7 and the properties of modular arithmetic. □

2 Lecture 2: Cryptography

Cryptography is concerned with creating algorithms to encode and decipher messages and information. It is impossible to overstate the importance of cryptography in everyday life: it is used in banking, online shopping, secure messaging and so much more.

Basic cryptographic methods have been around for thousands of years, but the field has evolved rapidly in the last century with the advent of computers. A famous example of an old method is the Caesar cipher, where each letter in the message is shifted by a fixed number of positions in the alphabet. More general permutations of the alphabet can be used as well of course, making the algorithm even more obscure. Methods like these ones are examples of **private key cryptography**, where the same key is used to encode and decode the message. The succes of the method relies on the secrecy of the algorithm and the key.

This is however not the most secure method of encryption, as the key must be shared between the sender and the receiver, but remain unknown to anyone else. To this end, modern algorithms often rely on a **public key system**, where the key used to encode the message is different from the key used to decode it. This means that a malevolent third party does not gain any advantage from knowing the encoding key, as it needs the decoding key as well to actually read the message.

2.1 RSA algorithm

One of the most famous public key algorithms is the **RSA algorithm**, named after its inventors Rivest, Shamir and Adleman. The RSA algorithm hinges on the fact that it is *complicated* to find solutions to equations of the form

$$x^e \equiv c \mod n\,. \tag{7}$$

SJS: The students can try to solve for $e = 7, c = 5, n = 143$. The answer is $x = 125$. Note that complicated here means that is computationally very intensive - with enough time and computing power, solutions to these equations can be found by brute-forcing all possible values of x.

In cryptography, one often talks about Alice and Bob, who want to communicate securely without the inference of a third party, Eve. For Alice and Bob to communicate securely by means of RSA, a couple steps are required, which are summarised below. The mathematical justification for these steps follows after.

First of all, Bob will choose two numbers for his public key, e and n. These two numbers have to satisfy a couple of conditions:

1. (ideally) n is the product of two large prime numbers, p and q.
2. The number e is coprime to $\phi(n)$, where ϕ is the Euler totient function.

Additionally, Bob will create a private key, d, which satisfies the equation

$$de \equiv 1 \mod \phi(n)\,. \tag{8}$$

Bob now publishes his public key, (e, n), and keeps his private key d secret.

When Alice wants to send a message to Bob, she will encode the message as a number M, e.g. by replacing letters by their number in the alphabet, or for more complicated messages by using ASCII. Alice splits her encoded message M in chunks M_i such that $M_i < n$, and encrypts them as $C_i = M_i^e \mod n$. She then sends the encrypted messages C_i to Bob.

For Eve to try and decipher these messages, she needs to solve the equation $x^e \equiv C_i \mod n$ for x, which is *complicated* as we mentioned before. For Bob, however, decoding is easier by means of his private key d. Indeed, as we explain the algroithm in more detail below, Bob recovers the original messages as

$$M_i \equiv C_i^d \mod n\,, \tag{9}$$

allowing him to read the messages Alice sent him.

2.1.1 Mathematical details

Let's start from the perspective of Bob, who wants to set up his public and private keys. The first thing he has to consider is the number n. This number will partially determine e and d satisfying $\gcd(e, \phi(n)) = 1$ and $de \equiv 1 \mod \phi(n)$. Importantly, he wants d to remain secret, and therefore it needs to be hard to calculate d. He can do this by making $\phi(n)$ hard to calculate, i.e. by making it hard to factorize n in primes.

It is known that factorizing large numbers is computationally expensive, and therefore calculating $\phi(n)$ is complicated for large numbers. However, Bob needs to know $\phi(n)$ himself! Checking whether a number is prime, however, is much easier. Therefore, Bob should construct a number n bottom-up, by fixing its prime factorization.

Exercise 3. *Should Bob fix a prime factorization with many prime numbers? What should the relative size of the prime numbers be? Keep Thm. 3 in mind.*

So, ideally Bob picks two large prime numbers p and q, in practice often having about 300 digits, and sets $n = pq$.

Exercise 4. *What is $\phi(pq)$ if p and q are prime numbers?*

Exercise 5. *Even though p, q should be of similar order of magnitude, they should still have a sufficiently large difference between them. Why is this?*

So, to summarize, Bob has chosen two large prime numbers p, q and set $n = pq$. He can now pick any number e coprime to $\phi(n)$ to form his public key. Using Thm. (4), there exist numbers $d, k \in \mathbb{Z}$ such that $de + k\phi(n) = 1$, i.e. $de \equiv 1 \mod \phi(n)$. Euclid's algorithm can be used to find these numbers.

Now it's Alice's turn to send a message to Bob. She encodes her message as a number M, and splits it in chunks M_i such that $M_i < n$. We take any such chunk M_i, and calculate

$$C_i = M_i^e \mod n\,.$$

This is straightforward to calculate, as the numbers e, n are known to Alice. The claim is now that Bob recovers the original messages as

$$M_i = C_i^d \mod n\,,$$

which is indeed the case. Note that $C_i^d \equiv (M_i^e)^d \equiv M_i^{de} \mod n$. We know that $de \equiv 1 \mod \phi(n)$, and therefore by 2 we have that $M_i^{de} \equiv M_i^1 \mod n$ IF M_i and n are coprime.

Exercise 6. *Does the result still hold if M_i and n are not coprime? Suppose $p|M_i$, and use Cor. 1.*

Exercise 7. *Why did we use that $M_i < n$?*

SJS: In class we will then do a cryptography battle. Divide the class in two teams, have both teams send a message from one half to the other half. First we check that the message is received and deciphered correctly. Then we pretend like the message was interecepted, and we see which team can break the other message first.

3 References

These notes are based on my own knowledge of these basic mathematical concepts, and the writing has been accelerated by the use of *Github copilot* and its implementation in VSCode. Inspiration has been taken from the course notes (particularly in Chapters 1 and 3) for *"Algebraische Structuren"* (Algebraic Structures), used in the first year of the Bachelor of Mathematics at the KU Leuven, at the time taught by Prof. Raf Cluckers - also the author of the lecture notes. The part on cryptography is largely taken from these notes as well.