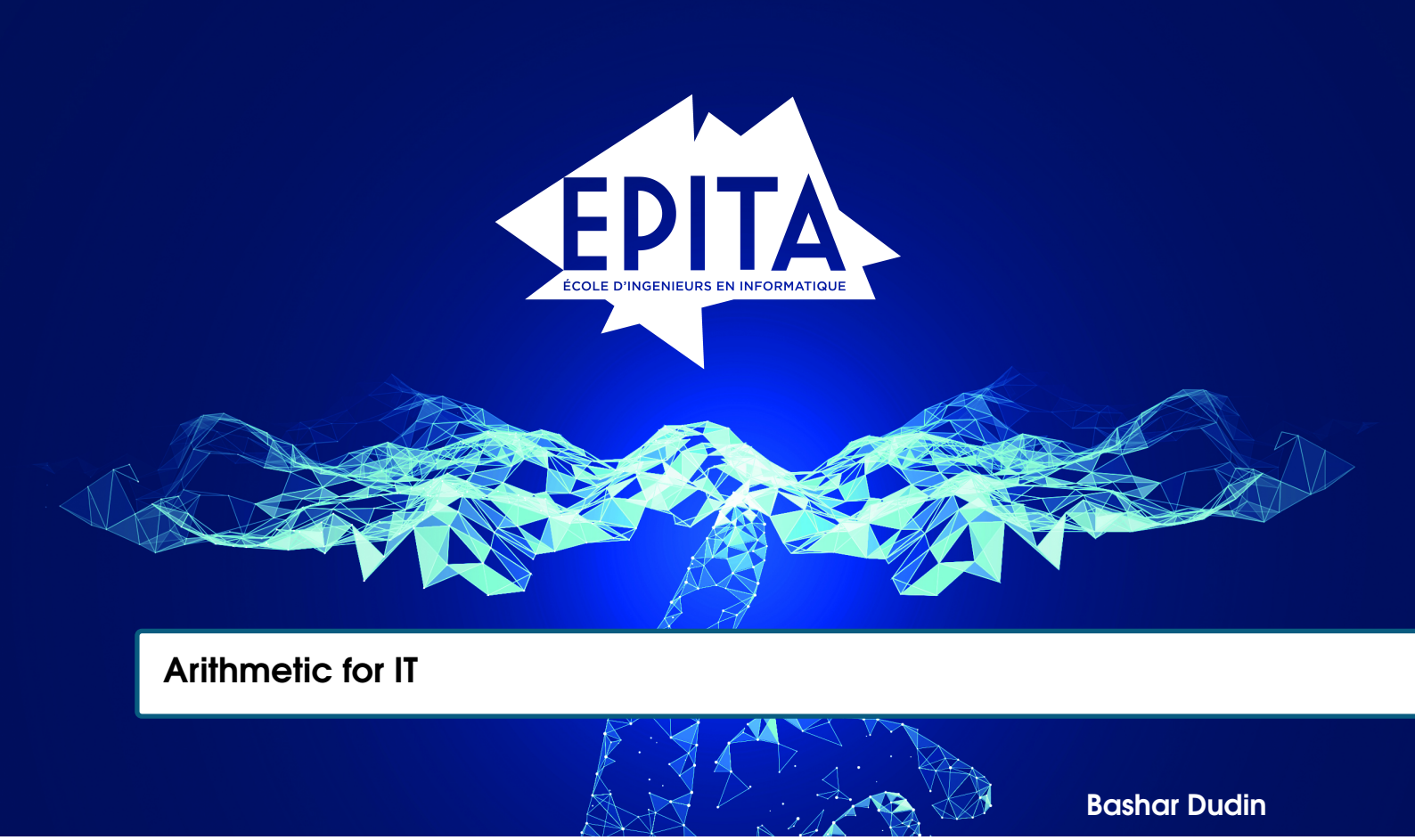

Arithmetic for IT

Bashar Dudin

Abstract

This is main reference for ***AFIT*** (Arithmetic for IT) project content. Project aims at generating RSA and ElGamal encryption data for educational purposes. On the way, students shall face core arithmetic notions needed to generate and manipulate such cryptosystems.

Contents

1 Introduction

> Arithmetic is a branch of mathematics that consists of the study of "numbers", especially the properties of the traditional operations on them – addition, subtraction, multiplication and division[1].

In integer arithmetic divisibility is one of the central notions we deal with : Are two given numbers multiple of each other? Are there specific numbers that don't have any non-trivial divisors? Can one decompose a given integer as a product of simpler (simplest) integer types from that perspective? You already know the answers to a number of these questions. Numbers that can only be divided by themselves or ± 1 are ***prime*** numbers. Any integer can be uniquely written as a product of prime numbers up to reordering.

These apparently simple questions are at the core of arithmetic uses in IT. Aside from the fact computers are integer calculators, divisibility questions are central in ciphering systems enabling secure exchange of information between two parties ; assuming a third party gets access to the exchanged information they will have too much trouble deciphering it to access the initial message.

The most known ciphering algorithm, ***RSA***[2], is based on the fact that factoring a number which is a product of two prime numbers is a hard question, needing much resources and time. On the opposite, computing powers of integers modulo a fixed given natural number is much less time-consuming.

The aim of this project is to get you through the arithmetic needed to try generating reasonably sized encryption systems. Do not be mistaken however ; ***we'll still be quite far from any real-life implementation of ciphering algorithms***.

2 How To Read This Document

This document is to serve as a mathematical reference for a coding project heavily involving elementary modular arithmetic. It contains examples, proofs and discussions needed to understand the core reason why used mathematical results are as stated and not otherwise. Everything you'll find here is there for a (good) reason ; the least of which is to put some sense into a number of apparently unexpected statements.

That being said an in-depth understanding of all aspects of this reference is not needed to complete a satisfying enough piece of the ***AFIT*** project. ***Do not hesitate to look for external documentation!*** ; this reference is here to serve as a structuring lattice for your own searches. If you find references easing your learning process : ***go for it!***

Here is a quick list of priorities to have, starting by high priority to low one. This is also a reading order you can stick to.

- Section 3 is core arithmetic knowledge and has to be fully mastered ; part of it is only rephrasing your in-class maths courses.
- Section 4 up till subsection 4.4 included is of essential importance to be able to write down your first cryptosystem.
- Section 6 is the main aim of ***AFIT***. Focus is on the ***RSA*** algorithm though, but both are expected to be implemented.
- Sections 4.5 and 5 are challenging, they should be left aside as long as a proper implementation of RSA and ElGamal cryptosystems hasn't been satisfyingly tested.

3 Integer Arithmetic

This section is a quick reminder about what you're going through in elementary arithmetic. `OCaml` primitives shall be given for basic arithmetic operations you're expected to use within your implementations.

Assumption 3.1 Our statements are going to be mainly focused on natural numbers. All have extensions to the case of integers, we shall not need them in that general setting.

3.1 Euclidean Division

[1] `https://en.wikipedia.org/wiki/Arithmetic`.
[2] Standing for its inventors' initials : *Rivest*, *Shamir* and *Adleman*.

Definition 3.1 Given a couple of natural numbers (n,p) the natural number p is said to divide n, denoted by $p \mid n$, if there is an integer $k \in \mathbb{N}$ such that $n = kp$.

R An integer n is said to be ***even*** if $2 \mid n$, it is ***odd*** otherwise.

R Any integer divides 0. Indeed, given an integer $n \in \mathbb{N}$ we can always write $0 = 0 \times n$.

If you take any two natural numbers randomly, there is little chance one of them divides the other. It is always possible to account for the lack of divisibility though ; this is enforced by what is called ***Euclidean division***[3].

Proposition 3.2 Given any couple of natural numbers $(a,b) \in \mathbb{N} \times \mathbb{N}^*$ then there is a unique couple (q,r) of natural numbers such that

$$a = bq + r \quad \text{where } 0 \leq r < b.$$

R The unique (q,r) has first entry called ***quotient*** of a by b while the second is called the ***remainder*** of the Euclidean division of a by b. The latter is often called ***remainder of*** a ***modulo*** b.

Proof. There are two statements in the previous proposition: one is about uniqueness and the other is about existence. Assuming there are two couples (q,r) and (q',r') satisfying statement then

$$bq + r = bq' + r' \Rightarrow b(q - q') = r' - r.$$

Now left hand of equality is in $\{-(b-1), b-1\}$ and is though a multiple of b. It has then to be 0. Thus $r = r'$ and then $q = q'$.
Existence is based on the following algorithmic procedure:

- If $0 \leq a < b$ then $(0,a)$ works
- Else add one to your quotient and look at Euclidean division of $a - b$ and b.

Such a procedure terminates because of a deep property of $\mathbb{N}$: any non-empty subset of $\mathbb{N}$ has a minimal element. The point is to show the set $\{q \mid a - bq < b\}$ is not empty. Intuition points out the fact it is the case, since left-hand side of condition could be as negative as we'd wish. This is indeed true but this stems from a deep fact we haven't shown. Knowing it is not empty, it has a minimal element q^*. For q^* the expression $a - bq^*$ can only be non-negative. Indeed, having q^* being the smallest element satisfying $a - bq^* < b$ then $a - b(q^* - 1) \geq b$. Notice the last inequality is obtained by adding b to the left-hand side. Both previous inequalities cannot happen at the same time if $a - bq^*$ is not non-negative, because otherwise adding b to the left-hand side wouldn't give anything bigger than b. □

In `OCaml` there is no primitive to compute Euclidean division at once. There are two inorder operators though to respectively compute quotient and remainder : `/` and `mod`.

R Using Euclidean division, the fact "$b \mid a$" is equivalent to the fact "remainder of Euclidean division of a by b is 0". Therefore, testing whether a number a is a multiple of b in `OCaml` is written

```
let is_divisible a b = (a mod b = 0) ;;
```

[3] Also called ***integer division***.

3.2 Primality

Definition 3.2 A natural number strictly bigger than 1 is said to be ***prime*** if it can only be divided by 1 and itself.

Checking that a number n is prime is a hard problem ; there is no other option but to go through the list of smaller natural numbers to check for divisibility. To be precise through natural numbers smaller than $\sqrt{n}$. Indeed, if $k \mid n$ then n/k does also divide n. Writing the couples $(k, n/k)$ of divisors of n one can figure out that at $\sqrt{n}$ one starts getting the same couples but with flipped entries. For instance for divisors of 36 we get

$$\begin{array}{cc} (1,36) & (36,1) \\ (2,18) & (18,2) \\ (4,9) & (9,4) \\ (6,6) & \end{array}$$

Importance of prime numbers comes from the following deep result:

Theorem 3.3 Any non-zero natural number n can be written as a product of prime numbers. This decomposition of n into a product of prime numbers is unique up to reordering of involved primes.

R A prime appearing in the decomposition of a natural number n is called a factor of n.

This theorem roughly says that knowing prime numbers is enough to understand all there is about natural numbers. The point is that generating prime numbers or characterising them is a highly challenging problem. An easier problem to take care of is the one of checking whether two given natural numbers have common factors.

Definition 3.3 Two non-zero natural numbers are said to be ***relatively prime*** or ***coprime*** if they don't have any common factors.

In order to tackle previous question of detecting whether two natural numbers are relatively prime we'll be introducing a new concept ; the GCD.

Definition 3.4 The GCD, short for Greater Common Divisor, of two non-zero integers a and b is the biggest integer d satisfying $d \mid a$ and $d \mid b$.

There is a point one needs to make clear here: why would such a maximal natural with such property exist?

- 1 does always satisfy this property which means that the set of natural numbers satisfying that property is not empty.
- Any such natural number is smaller that $\min\{|a|, |b|\}$, set is therefore bounded above. This gets us back to one of the core properties of $\mathbb{N}$ ensuring any non-empty subset bounded above has a maximal element.

N The GCD of two non-zero natural numbers a and b is denoted by $a \wedge b$.

Proposition 3.4 Two non-zero natural numbers a and b are relatively prime iff $a \wedge b = 1$.

3.3 Euclid's Algorithm

This algorithm is central for all arithmetic computational applications. It is the main reason why one can generate RSA public and private keys or parallelize integer computations.

The algorithm's idea is based on the following remark : let a and b be two non-zero natural numbers. Euclidean division gives a couple (q, r) of natural numbers such that

$$a = bq + r \qquad 0 \leq r < b. \tag{1}$$

If d divides a and b then it does divide $a - bq$; if $a = kd$ and $b = \ell d$ then

$$a - bq = kd - q\ell d = (k - q\ell)d.$$

Thus d divides r. This is true for any common divisor of a and b; this is specifically true for the GCD of a and b. Let us now assume that we'd go through this process iteratively : writing $r_0 = a$, $r_1 = b$, $q_1 = q$ and $r_2 = r$, equation (1) then becomes

$$r_0 = q_1 r_1 + r_2 \qquad 0 \leq r_2 < r_1 \tag{2}$$

where each divisor of r_0 and r_1 is also a divisor of r_2. Updating

$$\begin{array}{rcl} q_{n+1} & = & r_n / r_{n+1} \\ r_{n+2} & = & r_n \mod r_{n+1} \end{array}$$

we get the sequence of relations obtained through Euclidean division

$$\begin{array}{ccccccc} r_0 & = & q_1 r_1 & + & r_2 & & 0 \leq r_2 < r_1 \\ r_1 & = & q_2 r_2 & + & r_3 & & 0 \leq r_3 < r_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots \\ r_{n-1} & = & q_n r_n & + & r_{n+1} & & 0 \leq r_{n+1} < r_n \end{array}$$

At each level of the set of equations any common divisor of r_{n+1} and r_n is a divisor of r_{n+2}. At each such level the remainder r_n is an integer which is at least 1 less than the previous remainder unless it was already 0. All such remainders are non-negative, there is therefore a point after which all obtained remainders are 0. Let ℓ be the index of the last non-zero remainder in the previous sequence. We have then the sequence

$$\begin{array}{ccccccc} r_0 & = & q_1 r_1 & + & r_2 & & 0 \leq r_2 < r_1 \\ r_1 & = & q_2 r_2 & + & r_3 & & 0 \leq r_3 < r_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots \\ r_{n-1} & = & q_n r_n & + & r_{n+1} & & 0 \leq r_{n+1} < r_n \\ \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots \\ r_{\ell-2} & = & q_{\ell-1} r_{\ell-1} & + & r_\ell & & 0 \leq r_\ell < r_{\ell-1} \\ r_{\ell-1} & = & q_\ell r_\ell & & & & \end{array} \tag{3}$$

Let's look at these equations bottom up. The last non-zero remainder r_ℓ divides $r_{\ell-1}$. Now looking at previous line it does have to divide $r_{\ell-2}$. Going up through all equations we end up having a natural number dividing r_0 and r_1, i.e. dividing a and b. We thus get that $r_\ell \mid a \wedge b$. Going up down, any common divisor d of r_0 and r_1 has to divide r_ℓ, this is in particular the case of $a \wedge b$. We get that $a \wedge b \mid r_\ell$ and vice versa ; thus $r_\ell = a \wedge b$.

Proposition 3.5 The last non-zero remainder in the previous sequence is the GCD of first initial two terms.

R It is legitimate to wonder how many Euclidean divisions one needs to do at most to get the GCD of two given natural numbers. An easy bound to see is the one bounded by b. There is in fact a better bound than b, given by $2\log_2(b)+2$. You do not need to know how to figure it out ; it is enough to understand how quick a Euclidean division algorithm is.

R Deciding on relative primality of two given non-zero natural numbers is about executing the Euclidean division and getting 1 as last non-zero remainder.

3.4 Bézout Theorem

This section is devoted to the Bézout Theorem, claiming the existence of a Diophantine combination of two given integers equal to the GCD of these same two integers.

Theorem 3.6 Given a couple of non-zero natural numbers (a,b) there exists a couple of integers (u,v) such that

$$ua+vb = a\wedge b.$$

Proof. The proof is computational, it is based on Euclid's Algorithm. It is mainly about rewriting the series of equations 3. Expressing everything in terms of the remainders we get the equations

$$\begin{array}{ccccc} a & - & q_1 b & = & r_2 \\ b & - & q_2 r_2 & = & r_3 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ r_{\ell-2} & - & q_{\ell-1} r_{\ell-1} & = & a\wedge b \end{array} \tag{4}$$

Going bottom up, one can express each one of the remainders in terms of previous ones up till getting to a and b. This guess suggests that last relation is an integer combination of a and b. Here is a way to see it[4] in a proper way. Consider the following three equations, the last one is the first equation of previous relations 4.

$$\begin{array}{ccccc} 1\times a & - & 0\times b & = & r_0 \\ 0\times a & - & 1\times b & = & r_1 \\ 1\times a & - & q_1\times b & = & r_2 \end{array} \tag{5}$$

Notice the last equation is the first minus q_1 times the second. This pattern does in fact propagate till getting the GCD on the right-hand side of equalities. For instance using Euclidean division of r_1 by r_2 (given by second equation of 4) to subtract to second equation q_2 times the third in 5 we get

$$\begin{array}{ccccc} 1\times a & + & 0\times b & = & r_0 \\ 0\times a & + & (-1)\times b & = & r_1 \\ 1\times a & + & (-q_1)\times b & = & r_2 \\ (-q_2)\times a & + & (-1+q_1q_2)\times b & = & r_3 \end{array} \tag{6}$$

Going on step by step, using Euclidean division of successive remainders on the right-hand-sides to manipulate corresponding equations we get a relation between $a\wedge b$ and a, b defined by an integer combination of both. Writing (u_n) and (v_n) for the sequences of coefficients of a and b respectively the recursive definitions of both sequences are given by

$$\begin{cases} u_{n+1} & = & u_{n-1}-q_n u_n \\ v_{n+1} & = & v_{n-1}-q_n v_n \end{cases} \tag{7}$$

This is the starting point of a proper implementation of the extended Euclid Algorithm you're going to have to try out. □

Corollary 3.7 Two natural numbers a, b are relatively prime iff there is a couple of integers (u,v) such that

$$ua+vb = 1. \tag{8}$$

Proof. If a and b are relatively prime $a\wedge b = 1$. Following theorem 3.6 there is a couple (u,v) satisfying the expected relation. Now if there is a relation such as 8, then any divisor of a and b is also a divisor of 1. Since $a\wedge b$ is a positive divisor of 1 it has to be 1, thus a and b are relatively prime. □

We've already discussed (briefly) the complexity of Euclid's algorithm. It is time-wise the same for the extended Euclid's algorithm giving Bézout coefficients. We therefore ***do*** have an efficient algorithm to measure whether two natural numbers are relatively prime. That's only one of the applications of this algorithm, we'll be seeing a couple of others later on.

4 Modular Arithmetic

Integers are not the only mathematical objects one can do arithmetic with ; there are a series of these. Main mathematical objects for which one can talk about arithmetic are called ***rings***. The set $\mathbb{Z}$ of integers is only one case of such structures, there are many others. We shall be looking into extra examples of such objects, though we'll be only interested in basic computations with these ones.

[4] In fact to implement it!

4.1 Day of Week

Before getting into the core part of modular arithmetic, let's look at an example where such arithmetic appears : computing the day of the week a given date is.

> **Question 4-1** Assuming we're on Monday, how do you compute the day we'll be in 37 days from now?

A simple way to do so is to number the days of the week from 0 to 6 starting at the day you're at : Monday. Every 7 days you get back on Monday, that's something you know already. The Euclidean division of 37 by 7 is written

$$37 = 5 \times 7 + 2.$$

We thus get 5 times to Monday before going on to Wednesday whose number is 2. Thus the only number that matters in this question is `37 mod 7`. This argument is general: any number of days n after the first Monday one started with is going to get us back to `n mod 7`, which is a number between 0 and 6.

Let's add a more formal layer to get a better grasp on basic computations of week days. We're going to compute week days starting at the first day of our era following the Gregorian calendar. January, 1-st 0001 was a Saturday, let $\mathscr{W}$ denote the set of indices of week days, thus

$$\mathscr{W} = \{0,1,2,3,4,5,6\},$$

where 0 is a Saturday. We shall assume that number of days before and after January, 1-st 0001 is infinite[5].

> (R) We'll be calling ***date*** the number of days, before or after the first day of our era; this is to avoid ambiguity with the day of the week we're interested in computing.

Our main concern can be rephrased as:

> Given a date $d \in \mathbb{Z}$ what is the day of the week d corresponds to?

We've already answered above question previously: ***one only needs to look for the remainder of*** d ***modulo*** 7**.**

> (R) Be careful here about the fact that `OCaml mod` built-in function doesn't give the expected result if prefix (left) argument is negative. The `OCaml mod` function returns ***minus*** remainder of absolute value of prefix argument if the latter is negative. This ***does not*** follow in the integer division standard definition. The latter is the exact same definition you've seen for the case of natural numbers, thus remainder is always non-negative.

The computations involved in detecting week days of a given date involve a number of welcome compatibilities ; with respect to both addition and multiplication.

For instance, one could wonder if the 37-th day after Friday (6) is the same as the (`37 mod 7`)-th day after Friday. The day we're looking for is the 43-rd day after Saturday, one can write

$$43 = 6 \times 7 + 1$$

which gives a Sunday. We mainly did the following computation:

```
(37 + 6) mod 7.
```

Trying out the computation (which corresponds to the previously suggested one):

```
((37 mod 7) + (6 mod 7)) mod 7
```

we find back same result. This is a general fact. Let d_1 and d_2 be two given dates. Both have an integer division by 7 that can be written:

$$d_1 = 7 \times q_1 + r_1 \tag{9}$$
$$d_2 = 7 \times q_2 + r_2 \tag{10}$$

[5] Which is hardly conceivable going back and rather compromised going forward ...

Summing these two equations one gets:

$$(d_1+d_2) = 7\times(q_1+q_2)+(r_1+r_2).$$

There is no guarantee that (r_1+r_2) is smaller than 7. Looking into the integer division

$$(r_1+r_2) = 7\times s+t$$

We get that

$$(d_1+d_2) = 7\times(q_1+q_2+s)+t$$

with t being non-negative and smaller than 7. The last two relations assert that

$$(r_1+r_2) \mod 7 = (d_1+d_2) \mod 7$$

and that is exactly what we wrote down previously in our particular example.

Same type of compatibilities hold for multiplying dates. Assume we're looking at 3 times the 32-nd day. This is the 96-th day, it is given by `96 mod 7 = 5`, i.e. Thursday. This is the exact same result as the one given by

```
((3 mod 7) * (32 mod 7)) mod 7
```

This is a general fact as well ; reusing equations 9 one can write:

$$d_1d_2 = 7\times(7q_1q_2+q_2r_1+q_1r_2)+r_1r_2$$

without any guarantee on the fact r_1r_2 is non-negative and smaller than 7. Through the extra division

$$r_1r_2 = 7\times s+t$$

we figure out the Euclidean division

$$d_1d_2 = 7\times(7q_1q_2+q_2r_1+q_1r_2+s)+t$$

which exactly states that

$$d_1d_2 \mod 7 = r_1r_2 \mod 7.$$

To sum things up:

- The day of the week of a given date d is the remainder of the Euclidean division of d by 7.
- The day of the week of the sum of two dates d_1, d_2 is the remainder modulo 7 of the sum of both or equally the remainder of the sum of remainder of each.
- Previous bullet-point is also true in the case of multiplication. The remainder of multiplication of two dates d_1 and d_2 modulo 7 is the same as the remainder modulo 7 of multiplication of both remainders.

The computations we've met here are a basic manifestation of more general properties and constructions of central importance in arithmetic. They have a serious impact on integer programming within computers.

4.2 The Ring $\mathbb{Z}/n\mathbb{Z}$

We shall not define formally what a ring is. It is enough to know this is a set for which you have two binary operators called ***addition*** and ***multiplication*** that have the exact same properties you've always been using when dealing with integers. The ring we'll be defining next has an underlying ***finite*** set. This is of major importance when you're looking at things from a machine perspective ; anything that lives in such rings should be – up to memory issues – machine implementable.

Definition 4.1 Let $n>1$ be a positive integer. The ring $\mathbb{Z}/n\mathbb{Z}$ is the set

$$\mathbb{Z}/n\mathbb{Z} = \{0,\dots,n-1\}$$

together with the two binary operators $\oplus$ and $\otimes$ defined in the following fashion: Given any two elements $x, y \in \mathbb{Z}/n\mathbb{Z}$

$$x\oplus y = (x+y) \mod n \qquad \text{(addition)}$$

$$x \otimes y = (x \times y) \mod n. \qquad \text{(multiplication)}$$

Example 4.1 The simplest example is for $n = 2$. In that case $\mathbb{Z}/2\mathbb{Z} = \{0,1\}$. Addition and multiplication are simply given by the rules

$\oplus$	0	1
0	0	1
1	1	0

$\otimes$	0	1
0	0	0
1	0	1

Example 4.2 The case $n = 3$ is the set $\{0,1,2\}$ given by the addition and multiplication rules :

$\oplus$	0	1	2
0	0	1	2
1	1	2	0
2	2	0	1

$\otimes$	0	1	2
0	0	0	0
1	0	1	2
2	0	2	1

In practice we're often looking at the projection of integers modulo a given positive integer. This approach is represented by looking at the following simple map:

$$\begin{array}{rccc} \pi_n : & \mathbb{Z} & \longrightarrow & \mathbb{Z}/n\mathbb{Z} \\ & x & \longmapsto & x \mod n \end{array}$$

Example 4.3 The image of an element $x \in \mathbb{Z}$ by π_2 tells whether x is either odd or even. If $\pi_2(x) = 0$ then x is even otherwise it is odd. The image of x by π_3 being 0 says x can be divided by 3. If $\pi_3(x) = 1$ this means x is of the form $3k+1$ for some $k \in \mathbb{Z}$.

N It is standard to denote by $\bar{x}_n$ the quantity $\pi_n(x)$ or $x \mod n$. Depending on context the index might be left out as well.

Many central questions in modern arithmetic theory are summed up by :

> Let x be an integer whose remainders $\bar{x}_n$ modulo a reasonably big number of positive integers satisfy a property $\mathscr{P}$. Does x also satisfy $\mathscr{P}$?

Example 4.4 Let $\mathscr{P}$ be the property described by being smaller than 100. Let x be an integer, looking at $\bar{x}_n$ for $n \leq 100$ doesn't tell you anything about the fact $x \leq 100$. Indeed, any number has smaller remainder than 100 if taken modulo a smaller number than 100. Likewise if $\bar{x}_{101}$ is smaller than 100 that doesn't ensure x is. For instance $\overline{102}_{101} = 1$. Some meditation would get you to notice that if $x \leq 100$ then all remainders against integers $n \geq 101$ will always give you x back. The converse is also true. If all remainders $\bar{x}_n$ for $n \geq 101$ always give you x back then $x \leq 100$.

R The previous example is a dummy one, for a more accurate research question: ask!

The compatibilities we brought to light during section 4.1, regarding behaviour of addition and multiplication with respect to modulo operations, are general and expressed by: Given two integers $x, y \in \mathbb{Z}$ then

$$\overline{(x+y)}_n = \bar{x}_n \oplus \bar{y}_n \qquad \text{(addition)}$$

$$\overline{(xy)}_n = \bar{x}_n \otimes \bar{y}_n. \qquad \text{(multiplication)}$$

N In this line of work, notation is abused frequently. In many cases both $\bar{\cdot}$ and index are dropped. This is as well the case for $\oplus$ and $\otimes$ that are not standard notation, they are simply replaced by $+$ and $-$. In the following, we shall drop $\oplus$ and $\otimes$ all the time. Aside from this case we either keep $\bar{\cdot}$ and index or drop all but introduce the following notation: Given x, $y \in \mathbb{Z}$ then equality

$$\bar{x}_n = \bar{y}_n$$

shall be written as

$$x \equiv y\ [n]$$

or even sometimes as

$$x \equiv_n y.$$

Equality here is replaced by $\equiv$ and $[n]$ is to indicate the fact we're looking at remainders of x and y modulo n, or equivalently at $\bar{x}_n$ and $\bar{y}_n$ in $\mathbb{Z}/n\mathbb{Z}$. Previous compatibilities are thus written

$$x+y \equiv \bar{x}_n + \bar{y}_n\ [n] \qquad \text{(addition)}$$

$$xy \equiv \bar{x}_n\bar{y}_n\ [n] \qquad \text{(multiplication)}$$

4.3 Invertible Elements of $\mathbb{Z}/n\mathbb{Z}$

If you're looking at addition and multiplication for rational or real numbers, you know that for any non-zero number $x \in \mathbb{R}^*$ there is another number y such that $xy = 1$. For instance if $x = 2$ then $y = 0.5$. This is in general not the case anymore for $\mathbb{Z}/n\mathbb{Z}$. Here is the multiplication table for $\mathbb{Z}/4\mathbb{Z}$

$\otimes$	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	0	2
3	0	3	2	1

.

One can see that any element x except for 0 and 2 has a counterpart y such that $xy \equiv 1\ [n]$. The point is that 2 is non-zero but doesn't have any such counter-part, a behaviour that is different from what you're used to.

Definition 4.2 An element $x \in \mathbb{Z}/n\mathbb{Z}$ is said to be invertible if there is $y \in \mathbb{Z}/n\mathbb{Z}$ such that $xy \equiv 1\ [n]$.

The element y is ***unique*** and called the inverse of x in $\mathbb{Z}/n\mathbb{Z}$.

N The inverse of an invertible element $x \in \mathbb{Z}/n\mathbb{Z}$ is written x^{-1}. The set of invertible elements of $\mathbb{Z}/n\mathbb{Z}$ is denoted by $(\mathbb{Z}/n\mathbb{Z})^\times$. The number of invertible elements of $\mathbb{Z}/n\mathbb{Z}$ is written $\varphi(n)$; it is the cardinal of $(\mathbb{Z}/n\mathbb{Z})^\times$. In the literature $\varphi(n)$ is called the Euler number of n.

Let x be an invertible element in $\mathbb{Z}/n\mathbb{Z}$ whose inverse is y. By definition this means $xy \equiv 1\ [n]$; more explicitly there is $k \in \mathbb{Z}$ such that

$$xy + kn = 1. \tag{11}$$

Referring to Bézout Theorem this implies n and x are relatively prime. Conversely, if x and n are relatively prime there is a relation of type 11, modulo n this shows x has an inverse given by y.

Proposition 4.1 The set of invertible elements of $\mathbb{Z}/n\mathbb{Z}$ is equivalently the set of elements in $\{0,\dots,n-1\}$ relatively prime to n. This set is called the multiplicative group of $\mathbb{Z}/n\mathbb{Z}$.

Corollary 4.2 Given a prime number p all elements of $\{1,\dots,p-1\}$ are invertible, i.e.

$$(\mathbb{Z}/p\mathbb{Z})^\times = \{1,\dots,p-1\}.$$

Proof. Anyi nteger that is not a multiple of p is relatively prime to p. That is in particular the case of any non-zero element in $\mathbb{Z}/p\mathbb{Z}$. □

Writing down a function testing whether a given integer modulo n is invertible is about a proper use of the Bézout algorithm ; something we're already familiar with. One can ease the search a little though, by knowing a little more about the inner properties of elements in $(\mathbb{Z}/n\mathbb{Z})^\times$.

Proposition 4.3 Given two elements x, y that are invertible modulo n then xy is also invertible modulo n.

Proof. Let x^{-1} and y^{-1} be the respective inverses of x and y. The product $y^{-1}x^{-1}$ is then the inverse of xy. □

As soon as one finds an invertible element then all multiplicative powers of that element give other invertible elements.

Example 4.5 For example in the case of $\mathbb{Z}/5\mathbb{Z}$ the multiplicative group is $\{1,2,3,4\}$. The powers of 1 don't give much but 1. The powers of 2 modulo 5 span the set $\{1,2,3,4\}$.

We're not always as lucky as to find an integer whose powers span the whole multiplicative group, i.e. which enables us to recover all of its elements.

Example 4.6 In the case of $\mathbb{Z}/8\mathbb{Z}$ the multiplicative group is $\{1,3,5,7\}$. You can check that any element in $(\mathbb{Z}/8\mathbb{Z})^\times$ has square which is 1. Looking into powers of an invertible element x here doesn't give any other invertible element except for 1 if $x \neq 1$.

The subset of different elements one can generate by looking at powers of a given invertible element in the multiplicative group is of high interest in much of the arithmetic modulo n. Such subsets can measure the strength of an RSA private key. Valid public data for the ElGamal cryptosystem is an element that partially spans the multiplicative group of a specific $\mathbb{Z}/n\mathbb{Z}$. The next section is devoted to having a closer look at powers of invertible elements in $\mathbb{Z}/n\mathbb{Z}$.

4.4 Fermat's Little Theorem

Definition 4.3 Let x be an invertible element in $\mathbb{Z}/n\mathbb{Z}$ (thus an element in $(\mathbb{Z}/n\mathbb{Z})^\times$). The order of x is ***the*** smallest $k \in \mathbb{N}^*$ such that $x^k \equiv 1\ [n]$. We write $\mathrm{ord}_n(x)$ for the order of x in n.

Example 4.7 In the case $n = 8$ the invertible elements of $\mathbb{Z}/8\mathbb{Z}$ are 1, 3, 5, 7. The first is of order 1 the latter of order 2.

Example 4.8 The multiplicative group of $\mathbb{Z}/9\mathbb{Z}$ is given by the elements 1, 2, 4, 5, 7 and 8. Respectively of orders 1, 6, 3, 6, 3 and 2.

The emphasized definite article in definition 4.3 suggests there is always one such smallest positive integer being the order of x. This implicitly expresses the fact the set $\{k \mid x^k \equiv 1\ [n]\}$ is not empty. Though we tested this fact on two simple examples, we've shown no guarantee this is the case in general until now.

Theorem 4.4 Let x be an invertible element in $\mathbb{Z}/n\mathbb{Z}$. Then $x^{\varphi(n)} \equiv 1\ [n]$.

Proof. To ease notation we write G_n for the multiplicative group $(\mathbb{Z}/n\mathbb{Z})^\times$. The proof we're giving here is based on a very structural understanding of how an element of G_n acts on its environment. Let m_x be the map $m_x : G_n \to G_n$ sending an element $y \in G_n$ on xy. Taking the example when $n = 9$ and $x = 2$ the map m_2 has domain and target $G_9 = \{1,2,4,5,7,8\}$. It sends the list of elements $[1;2;4;5;7;8]$ entry-wise to $[2;4;8;1;5;7]$. Thus sends

$$\begin{array}{ccc} 1 & \to & 2 \\ 2 & \to & 4 \\ 4 & \to & 8 \\ 5 & \to & 1 \\ 7 & \to & 5 \\ 8 & \to & 7 \end{array}$$

You can check that this defines a bijection of G_9 on itself ; this is something we call a permutation. The image of m_2 is equal here to G_9. This observation is much more general: m_x is a bijection from G_n on itself.

To show m_x is injective assume there are two elements y_1 and y_2 in G_n such that $m_x(y_1) = m_x(y_2)$. This means

$$xy_1 \equiv xy_2\ [n].$$

By definition x is invertible, multiplying previous relation by x^{-1} gets $y_1 \equiv y_2\ [n]$. To check m_x is surjective one can see that given any element y in G_n the element $t = x^{-1}y$ in G_n satisfies $m_x(t) = y$.

The fact that m_x is bijective implies that we have an equality between the sets

$$\{xy \mid y \in G_n\} = \{y \mid y \in G_n\}$$

the product of all elements of right-hand and left-hand sets are equal (we have the same sets on both sides). This is written

$$x^{\varphi(n)}\Big(\prod_{y\in G_n} y\Big) \equiv \Big(\prod_{y\in G_n} y\Big)\ [n].$$

The product of invertible elements is invertible, multiplying previous relation by its inverse we get

$$x^{\varphi(n)} \equiv 1\ [n],$$

which is what we expect. □

Corollary 4.5(Fermat's Little Theorem Let p be a prime number, a let x be a non-zero element in $\mathbb{Z}/p\mathbb{Z}$, then

$$x^{p-1} \equiv 1\ [p].$$

Proof. The set of invertible element of $\mathbb{Z}/p\mathbb{Z}$ is exactly the set of its non-zero elements and $\varphi(p) = p-1$. □

R Fermat's Little Theorem is also stated as : given any element x in $\mathbb{Z}/p\mathbb{Z}$ we have that $x^p \equiv x\ [p]$. It is equivalent to previous statement. If x is invertible multiplying equation by x^{-1} gives back 4.5. If x is not, it is zero and relation just says $0 \equiv 0\ [p]$ which is indeed true.

In the two examples 4.7 and 4.8 we have $\varphi(8) = 4$ and $\varphi(9) = 6$. The orders of invertible elements in $\mathbb{Z}/8\mathbb{Z}$ are all divisors of $\varphi(8)$. That is the case as well for orders of invertibles of $\mathbb{Z}/9\mathbb{Z}$. This fact is general.

Proposition 4.6 Let x be an invertible element in $\mathbb{Z}/n\mathbb{Z}$. An element $m \in \mathbb{N}^*$ satisfies $x^m \equiv 1\ [n]$ if and only if it is a multiple of $\mathrm{ord}_n(x)$.

Proof. Let k be the order of x modulo n. The Euclidean division of m by k gives the relation $m = kq + r$ where $0 \leq r < k$. We thus get

$$x^m \equiv x^{kq}x^r\ [n] \tag{12}$$
$$1 \equiv x^r\ [n]. \tag{13}$$

If r is positive then r would satisfy $x^r \equiv 1\ [n]$ and be smaller than k, which is not possible by definition of k (the smallest positive integer satisfying $x^k \equiv 1\ [n]$). Then $r = 0$ and m is indeed a multiple of the order of x. □

Corollary 4.7 The order of an invertible element in $\mathbb{Z}/n\mathbb{Z}$ divides $\varphi(n)$.

Proof. This is due to the fact $x^{\varphi(n)} \equiv 1\ [n]$ following 4.4. □

4.5 Chinese Remainder Theorem

It is a standard practice in mathematics to try understanding an object by identifying it to a composite of easier-to-understand sub-objects. This is also a philosophy one en s in computer science : this is more or less the principle of "divide and conquer" strategies. Not to mention the fact software is mainly thought of as a series of components linked together and each devoted to a given task. In the case of modular arithmetic one can decompose many $\mathbb{Z}/n\mathbb{Z}$ into cartesian products of smaller $\mathbb{Z}/m\mathbb{Z}$ sets. This is what we'll be going through in this section.

Let m and n be two relatively prime integers bigger than 1. Let ψ be the map

$$\begin{array}{rccc} \psi: & \mathbb{Z}/nm\mathbb{Z} & \longrightarrow & \mathbb{Z}/n\mathbb{Z}\times\mathbb{Z}/m\mathbb{Z} \\ & x & \longmapsto & (\bar{x}_n, \bar{x}_m) \end{array}$$

keeping the remainder of $x \in \{0,\dots,nm-1\}$ modulo n and modulo m respectively as first and second entry of a couple in $\mathbb{Z}/n\mathbb{Z}\times\mathbb{Z}/m\mathbb{Z}$.

Example 4.9 Let's consider the case $(n,m) = (2,3)$. The map ψ is one having domain $\mathbb{Z}/6\mathbb{Z}$ and target $\mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/3\mathbb{Z}$. Here are the list of images of the 6 elements of $\mathbb{Z}/6\mathbb{Z}$ by ψ:

$$\begin{array}{ccc} 0 & \to & (0,0) \\ 1 & \to & (1,1) \\ 2 & \to & (0,2) \\ 3 & \to & (1,0) \\ 4 & \to & (0,1) \\ 5 & \to & (1,2) \end{array}$$

You can first notice that this map is bijective. Thus $\mathbb{Z}/6\mathbb{Z}$ contains as much information as $\mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/3\mathbb{Z}$. There is in fact more: the arithmetic operations on $\mathbb{Z}/6\mathbb{Z}$ can be reflected on $\mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/3\mathbb{Z}$ through ψ. Let us define addition and multiplication operators on $\mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/3\mathbb{Z}$ in the following fashion: given (x_1,x_2) and $(y_1,y_2) \in \mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/3\mathbb{Z}$ we have

$$\begin{aligned} (x_1,x_2) + (y_1,y_2) &= (x_1+y_1, x_2+y_2) && \text{(addition)} \\ (x_1,x_2) \times (y_1,y_2) &= (x_1y_1, x_2y_2). && \text{(multiplication)} \end{aligned}$$

These are called component-wise addition and multiplication. Here are the addition and multiplication tables for $\mathbb{Z}/6\mathbb{Z}$ with usual operations and $\mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/3\mathbb{Z}$ with the operations we just defined.

$\mathbb{Z}/6\mathbb{Z}$

$+$	0	1	2	3	4	5
0	0	1	2	3	4	5
1	1	2	3	4	5	0
2	2	3	4	5	0	1
3	3	4	5	0	1	2
4	4	5	0	1	2	3
5	5	0	1	2	3	4

$\times$	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	1	2	3	4	5
2	0	2	4	0	2	4
3	0	3	0	3	0	3
4	0	4	2	0	4	2
5	0	5	4	3	2	1

$\mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/3\mathbb{Z}$

$+$	$(0,0)$	$(1,1)$	$(0,2)$	$(1,0)$	$(0,1)$	$(1,2)$
$(0,0)$	$(0,0)$	$(1,1)$	$(0,2)$	$(1,0)$	$(0,1)$	$(1,2)$
$(1,1)$	$(1,1)$	$(0,2)$	$(1,0)$	$(0,1)$	$(1,2)$	$(0,0)$
$(0,2)$	$(0,2)$	$(1,0)$	$(0,1)$	$(1,2)$	$(0,0)$	$(1,1)$
$(1,0)$	$(1,0)$	$(0,1)$	$(1,2)$	$(0,0)$	$(1,1)$	$(0,2)$
$(0,1)$	$(0,1)$	$(1,2)$	$(0,0)$	$(1,1)$	$(0,2)$	$(1,0)$
$(1,2)$	$(1,2)$	$(0,0)$	$(1,1)$	$(0,2)$	$(1,0)$	$(0,1)$

$\times$	$(0,0)$	$(1,1)$	$(0,2)$	$(1,0)$	$(0,1)$	$(1,2)$
$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$
$(1,1)$	$(0,0)$	$(1,1)$	$(0,2)$	$(1,0)$	$(0,1)$	$(1,2)$
$(0,2)$	$(0,0)$	$(0,2)$	$(0,1)$	$(0,0)$	$(0,2)$	$(0,1)$
$(1,0)$	$(0,0)$	$(1,0)$	$(0,0)$	$(1,0)$	$(0,0)$	$(1,0)$
$(0,1)$	$(0,0)$	$(0,1)$	$(0,2)$	$(0,0)$	$(0,1)$	$(0,2)$
$(1,2)$	$(0,0)$	$(1,2)$	$(0,1)$	$(1,0)$	$(0,2)$	$(1,1)$

We have placed an element $x \in \mathbb{Z}/6\mathbb{Z}$ and its image $\psi(x) \in \mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/3\mathbb{Z}$ at same position of respective tables. Looking into it you can check the two following facts: for every $x, y \in \mathbb{Z}/6\mathbb{Z}$ we have

$$\psi(x+y) = \psi(x) + \psi(y) \qquad \psi(x \times y) = \psi(x) \times \psi(y).$$

To sum things up, ψ is here a bijection transforming arithmetic operations on domain into compatible ones on the target. One can either make computations on the right column then go left or the other way around.

The previous example is only an instance of what happens in general. This is summed up in the Chinese Remainder Theorem we're stating now.

Theorem 4.8 Let n, m be two positive integers $n, m > 1$ that are relatively prime. The map

$$\begin{array}{rccc} \psi: & \mathbb{Z}/nm\mathbb{Z} & \longrightarrow & \mathbb{Z}/n\mathbb{Z} \times \mathbb{Z}/m\mathbb{Z} \\ & x & \longmapsto & (\bar{x}_n, \bar{x}_m) \end{array}$$

is a bijective map such that for each $x, y \in \mathbb{Z}/nm\mathbb{Z}$ we have the compatibilities

$$\psi(x+y) = \psi(x) + \psi(y) \quad \text{(addition)}$$
$$\psi(x \times y) = \psi(x) \times \psi(y). \quad \text{(multiplication)}$$

Be careful about the fact that arithmetic operations in above compatibilities are not defined in the same fashion on both sides of equality. The left part is addition as we understand it in $\mathbb{Z}/nm\mathbb{Z}$, the second being component-wise in $\mathbb{Z}/n\mathbb{Z} \times \mathbb{Z}/m\mathbb{Z}$.

Proof. Addition and multiplication compatibilities are independent from the fact that ψ would be a bijection. Taking two elements $x, y \in \mathbb{Z}/nm\mathbb{Z}$ then, by definition

$$\psi(x+y) = (\overline{(x+y)}_n, \overline{(x+y)}_m) \tag{14}$$

using compatibility of modulo to addition

$$\psi(x+y) = (\bar{x}_n + \bar{y}_n, \bar{x}_m + \bar{y}_m) \tag{15}$$

by definition of addition in $\mathbb{Z}/n\mathbb{Z} \times \mathbb{Z}/m\mathbb{Z}$

$$\psi(x+y) = (\bar{x}_n, \bar{y}_m) + (\bar{x}_n, \bar{y}_m) \tag{16}$$

lastly, by definition of ψ

$$\psi(x+y) = \psi(x) + \psi(y) \tag{17}$$

The exact same reasoning applied to multiplication gives

$$\psi(x \times y) = \psi(x) \times \psi(y).$$

Let's come to work on the surjective and injective aspects. The core reason for why ψ is both surjective and injective is the relative primality of n and m. This result does in fact extend to much broader contexts.

The map ψ is injective.
Assume there are two elements $x, y \in \mathbb{Z}/nm\mathbb{Z}$ having same image by ψ, i.e. $\psi(x) = \psi(y)$. This does equivalently mean that

$$\begin{aligned} \bar{x}_n &\equiv \bar{y}_n \ [n] \\ \bar{x}_m &\equiv \bar{y}_m \ [m] \end{aligned}$$

Thus

$$\begin{aligned} \overline{(x-y)}_n &\equiv 0 \ [n] \\ \overline{(x-y)}_m &\equiv 0 \ [m] \end{aligned}$$

and both n and m divide $x-y$. The Smallest Common Multiple of n and m does then divide $x-y$. Since $n \wedge m = 1$, that SCM is nm[6]. We thus get $x \equiv y \ [nm]$ which is what we expect.

The map ψ is surjective.
The surjective aspect can be deduced from the fact both domain and target have same cardinal and the map is injective. As is always the case with any computer scientist we're interested in an explicit construction of inverse of ψ : a function ϕ sending $y = (y_1, y_2) \in \mathbb{Z}/n\mathbb{Z} \times \mathbb{Z}/m\mathbb{Z}$ on $x \in \mathbb{Z}/nm\mathbb{Z}$ such that $\psi(x) = y$. By hypothesis there are integers u and v such that

$$un + vm = 1.$$

Looking at this relation both modulo n and m we get that

$$\begin{aligned} vm &\equiv 1 \quad [n] \\ un &\equiv 1 \quad [m] \end{aligned}$$

[6] Gauss Lemma!

The integer $x = y_1vm + y_2un$ does then satisfy

$$\begin{array}{rcccl} y_1vm + y_2un & \equiv & y_1vm & \equiv & y_1\ [n] \\ y_1vm + y_2un & \equiv & y_2un & \equiv & y_2\ [m] \end{array}$$

which is exactly saying $\psi(x) = y$. Thus ϕ is defined by

$$\phi(y_1, y_2) = y_1vm + y_2un\ [nm]$$

where v and u are previously defined integers coming from the Bézout relation, $y_1 \in \{0, \ldots, n-1\}$ and $y_2 \in \{0, \ldots, m-1\}$. To be accurate one would need to check that any other choice (u', v') of Bézout coefficients would give the same map ϕ. Any such other choice can be written as $(u', v') = (u + km, v - kn)$ for some $k \in \mathbb{Z}$. But then the expression becomes

$$\begin{aligned} y_1v'm + y_2u'n &= y_1(v - kn)m + y_2(u + km)n && (18) \\ &= y_1v + y_2m + (-y_1k + y_2k)mn && (19) \\ &\equiv y_1v + y_2m\ [nm]. && (20) \end{aligned}$$

The choice of different Bézout coefficients gives indeed same inverse ϕ. □

Example 4.10 Let's look at a quick example to check for strategy to build up inverse element by ψ. Take $(n, m) = (4, 7)$ and consider the couple $(2, 5) \in \mathbb{Z}/4\mathbb{Z} \times \mathbb{Z}/7\mathbb{Z}$. Following steps of previous proof we first need to have Bézout coefficients assessing the fact 4 and 7 are relatively prime. A quick computation gives

$$(-5) \times 4 + 3 \times 7 = 1.^a$$

The element

$$5 \times (-5) \times 4 + 2 \times 3 \times 7 \equiv 26\ [28]$$

has remainder modulo 4 which is 2 and one modulo 7 which is 5, i.e. $\psi(26) = (2, 5)$.

[a] These are not the only Bézout coefficients but the ones given by Euclid's algorithm.

Corollary 4.9 Given an integer $k > 1$, let $m_1, \ldots, m_k$ be a list of positive integers > 1 that are pairwise relatively prime. Denote by m the product of m_1 up to m_k. The map

$$\begin{array}{rccc} \psi: & \mathbb{Z}/m\mathbb{Z} & \longrightarrow & \mathbb{Z}/m_1\mathbb{Z} \times \cdots \times \mathbb{Z}/m_k\mathbb{Z} \\ & x & \longmapsto & (\bar{x}_{m_1}, \ldots, \bar{x}_{m_k}) \end{array}$$

is a bijection that is compatible to component-wise addition and multiplication on target.

Proof. We'll be mainly giving outlines of proof in the following. Formalizing it properly would take us astray.

Compatibility to pairwise addition and multiplication is tautological and stems from the fact modulo operation is compatible to both addition and multiplication. Proof of bijective aspect is algorithmic ; it is based on an inductive use of the Chinese Theorem approach for building up an inverse map.

If we wanted to check map is injective we find ourselves in the exact same position as was the case in the proof of 4.8. Two elements that have equal image have a difference that can be divided by m_1, …, m_k. Since these are pairwise relatively prime it divides m. Thus equality modulo m. This is again enough to show map is a bijection. The point being that both domain and target have same cardinal.

Here is how building up inverse image of an element $(y_1, \ldots, y_k)$ goes : assume you have an inverse image $y_{1\cdots j}$ of $(y_1, \ldots, y_j)$ for $1 \leq j < k$ in $\mathbb{Z}/(m_1 \cdots m_j)\mathbb{Z}$. Then an inverse image of $(y_1, \ldots, y_j, y_{j+1})$ in $\mathbb{Z}/(m_1 \cdots m_{j+1})\mathbb{Z}$ is

$$y_{1\cdots j}um_{j+1} + y_{j+1}v(m_1 \cdots m_j)$$

where u and v are Bézout coefficients respectively for m_{j+1} and $m_1 \cdots m_j$. □

5 Input of the CRT

Core questions in arithmetic are about identifying prime and invertible numbers. Aim is then to be able to factor a given non-invertible number into its prime components. Factoring is a heavy process ; the reason why RSA ciphering method

is secure enough when having high enough prime numbers involved. Another type of difficulty is related to the fact that looking at powers of big enough integers has a high toll on computations. Many standard ciphering methods are based on taking powers of such high enough numbers.

The Chinese Remainder Theorem can help out breaking into smaller problems each of the previous issues. This is already something we can imagine for the case of powers of an integer:
Let's assume we're working with integers smaller than a fixed given integer M. Let

$$M = M_1 M_2 \cdots M_h$$

where M_js are pairwise relatively prime integers. The CRT states that

$$\mathbb{Z}/M\mathbb{Z} \simeq \mathbb{Z}/M_1\mathbb{Z} \times \cdots \times \mathbb{Z}/M_h\mathbb{Z}$$

where $\simeq$ is a notation to specify that there is a bijective map from one to the other, which is compatible to arithmetic operations ; this is what we call an ***isomorphism***. Given non-negative integers x and k such that x^k is smaller than M then one can compute x^k by looking at the k-power of each component of

$$\left(\bar{x}_{M_1}, \bar{x}_{M_2}, \ldots, \bar{x}_{M_h}\right)$$

then building up inverse image through the CRT map. The fact each single computation is quicker on the right-hand side is related to the fact that factors are smaller, since they're constants of the system one might have computed their Euler numbers and thus make use of simplifications (integer divisions) to compute these powers.

R Notice that if M is chosen prime or power of a prime, such strategy does not make sense. The CRT doesn't give any decomposition in this case.

The previous approach is the standard one for the use of the CRT: We're willing to make a specific check or computation in $\mathbb{Z}/M\mathbb{Z}$, we look at image of data in $\mathbb{Z}/M_1\mathbb{Z} \times \cdots \mathbb{Z}/M_h\mathbb{Z}$ make quicker equivalent computations component-wise then get back output results on $\mathbb{Z}/M\mathbb{Z}$.

5.1 Computing Invertibles

Let's look back at the previous example of $\mathbb{Z}/6\mathbb{Z}$. Following the CRT we have an isomorphism between $\mathbb{Z}/6\mathbb{Z}$ and $\mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/3\mathbb{Z}$ given by the map the output of which is composed of reduction modulo 2 and 3 of input.

$\mathbb{Z}/6\mathbb{Z}$
0
1
2
3
4
5

$\mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/3\mathbb{Z}$

	0	1
0	(0,0)	(0,1)
1	(1,0)	(1,1)
2	(2,0)	(2,1)

The invertible elements in $\mathbb{Z}/6\mathbb{Z}$ are the shaded ones. Their image in $\mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/3\mathbb{Z}$ are the shaded cells corresponding to $(1,1)$ and $(1,2)$. In $\mathbb{Z}/6\mathbb{Z}$ the inverse of 1 is itself and so is the case for 5. Multiplying $(1,1)$ and $(1,2)$ each one by itself we get $(1,1)$ for first and $(1,4) = (1,1)$ for second. Thus, for each element x of either of them there is an element y such that $xy = (1,1)$. In a sense we're saying that both elements $(1,1)$ and $(1,2)$ are invertible in $\mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/3\mathbb{Z}$. The element $(1,1)$ replaces in this context the element $1 \in \mathbb{Z}/6\mathbb{Z}$. It is called the neutral element of $\mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/3\mathbb{Z}$ since multiplication of any element $z \in \mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/3\mathbb{Z}$ by $(1,1)$ gives back z. Rephrasing this remark one can say that invertible elements in $\mathbb{Z}/6\mathbb{Z}$ correspond to invertible ones in $\mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/3\mathbb{Z}$.

The element of a product $\mathbb{Z}/M_1\mathbb{Z} \times \cdots \times \mathbb{Z}/M_h\mathbb{Z}$ only composed of entries equal to 1 is called the neutral element of that product. The multiplication of any element z by $(1,\ldots,1)$ is equal to z. An element $x \in \mathbb{Z}/M_1\mathbb{Z} \times \cdots \times \mathbb{Z}/M_h\mathbb{Z}$ is said to be invertible if there is an element y such that $xy = (1,\ldots,1)$. These definitions generalize the ones we've seen for $\mathbb{Z}/n\mathbb{Z}$.

Proposition 5.1 An element $x \in \mathbb{Z}/M\mathbb{Z}$ is invertible iff its image $\psi(x)$ by the CRT map is so.

Proof. Let x be an invertible element in $\mathbb{Z}/M\mathbb{Z}$. By definition, there is $y \in \mathbb{Z}/M\mathbb{Z}$ such that $xy \equiv 1\ [M]$. Taking image by the CRT map ψ of both hand-sides of equation we get

$$\psi(xy) = \psi(x)\psi(y) = (1,\dots,1)$$

which does exactly say that any invertible element x of $\mathbb{Z}/M\mathbb{Z}$ is sent on an invertible one of $\mathbb{Z}/M_1\mathbb{Z} \times \cdots \times \mathbb{Z}/M_h\mathbb{Z}$. Conversely let x be an element in $\mathbb{Z}/M\mathbb{Z}$ such that $\psi(x)$ is invertible in $\mathbb{Z}/M_1\mathbb{Z} \times \cdots \times \mathbb{Z}/M_h\mathbb{Z}$. By definition there is $\bar{y} \in \mathbb{Z}/M_1 \times \cdots \times \mathbb{Z}/M_h\mathbb{Z}$ such that $\psi(x)\bar{y} = (1,\dots,1)$. Since ψ is an isomorphism, there is $y \in \mathbb{Z}/M\mathbb{Z}$ such that $\psi(y) = \bar{y}$. We can thus write that

$$\psi(1) = (1,\dots,1) = \psi(x)\psi(y) = \psi(xy).$$

Applying ψ^{-1} at extremities of the above sequence of equalities we get $xy \equiv 1\ [M]$. □

Corollary 5.2 The CRT map ψ defines a bijection

$$\psi^{\times} : (\mathbb{Z}/M\mathbb{Z})^{\times} \longrightarrow (\mathbb{Z}/M_1 \times \cdots \times \mathbb{Z}/M_h\mathbb{Z})^{\times}$$

compatible to multiplication. It is defined as the h-tuple of the reductions modulo M_i of the input ; as is the case for ψ.

In order to compute inverse of element x in $\mathbb{Z}/M\mathbb{Z}$ one can

- compute image $(\bar{x}_1,\dots,\bar{x}_h)$ of x by the CRT map in $\mathbb{Z}/M_1 \times \cdots \times \mathbb{Z}/M_h\mathbb{Z}$;
- find inverse $\bar{y}_i$ of each element $\bar{x}_i$ in $\mathbb{Z}/M_i\mathbb{Z}$ if any;
- if previous step doesn't go through x wasn't invertible, otherwise compute inverse y by CRT map of $(\bar{y}_1,\dots,\bar{y}_h)$;
- computed y is inverse of x.

5.2 Factoring Integers

With a little more care, the previous divide and conquer strategy might be fruitful to factor integers smaller than M. Let's first check the simple example $M = 15$. The CRT claims the CRT map ψ gives an isomorphism

$$\mathbb{Z}/15\mathbb{Z} \simeq \mathbb{Z}/3\mathbb{Z} \times \mathbb{Z}/5\mathbb{Z}.$$

On the left-hand side of isomorphism we get the decomposition $14 = 2 \times 7$. Taking image of both sides of inequality by ψ we get the equation $(2,4) = (2,2) \times (1,2)$. Which one can still write as

$$\psi(14) = (2,4) = (2 \times 1, 2 \times 2) = (2,2) \times (1,2) = \psi(2) \times \psi(7).$$

It does suggests the idea that factoring 14 is about factoring each one of the entries of its image under the CRT map, then rebuilding inverse images of each factor using the CRT map. For this to make sense one would first need each factor to have an inverse image by the CRT map that is smaller than the integer we're expecting to factor. For instance if you take 12 its image in $\mathbb{Z}/3\mathbb{Z} \times \mathbb{Z}/5\mathbb{Z}$ is $(0,2)$ which, for instance, is the product of $(0,4)$ and $(0,3)$. Now $(0,4)$ has inverse image by the CRT map which is 9 and $(0,3)$ has such inverse given by 3. The product of 9 by 3 does give 12 only modulo 15 and is not an integer decomposition of 12. Notice that the previous decomposition of $(0,2)$ is not unique, another decomposition could have given a different output. An even more striking example is the one taken with 5. Its image by under the CRT map is $(2,0)$ whose the product of $(2,0)$ and $(1,0)$. Their respective images are 5 and 10, the product of which is indeed 5 modulo 15, but certainly not a non-trivial decomposition of the ***prime*** 5 in $\mathbb{Z}$.

Rather than giving a complete understanding of this phenomenon as is, let's have a look at it through the forceful search for factors. Let's consider we're looking for a factor of an integer m smaller than M. One naive but valid way to do so is to go through all integers from 2 up to $\sqrt{m}$, testing for divisibility. Equivalently we can go through integers k starting at 2 and adding 1 till we either find a divisor of m or get $k^2 > m$. On the right-hand of the CRT map, i.e. in $\mathbb{Z}/M_1\mathbb{Z} \times \cdots \mathbb{Z}/M_h\mathbb{Z}$, this is equivalently given by:

1. Let $\kappa = (\bar{2}_{M_1},\dots,\bar{2}_{M_h})$.
2. Loop as long as κ has a component whose square is bigger than corresponding component of $\psi(m)$.
3. Test whether κ divides $\psi(m)$ component-wise;

- if not increment κ by $(1,\dots,1)$ and loop back
- else inverse image k of κ by the CRT map is a factor of m.

R In case m is invertible and can be factored by k then co-factor of m by k is mk^{-1}. Otherwise there is no unique co-factor. This is the case above for $12 \equiv 3\times 9 \equiv 3\times 4$ [15].

6 Ciphering : An *Ersatz*

Ciphering is the act of transforming a message into an unintelligible one except for the recipient. It does involve a ciphering procedure and a deciphering one. From an abstract perspective, letting $\mathscr{M}$ be the set of messages and $\mathscr{C}$ the set of encrypted messages, a ciphering method involves two maps $\mathfrak{c} : \mathscr{M} \to \mathscr{C}$ and $\mathfrak{d} : \mathscr{C} \to \mathscr{M}$ such that $\mathfrak{d}\circ\mathfrak{c} = \mathrm{id}_{\mathscr{M}}$. The sender has to know of $\mathfrak{c}$ and the recipient of $\mathfrak{d}$. There are a couple of properties we expect from a cipher:

- Images of the methods $\mathfrak{c}$ and $\mathfrak{d}$ are easy and quick to compute;
- The methods $\mathfrak{c}$ and $\mathfrak{d}$ are hard to figure out if you're only having a subset (all) of encrypted messages $\mathscr{C}$.

R Everything in a computer being sequences of numbers the sets of messages $\mathscr{M}$ and encrypted ones $\mathscr{C}$ shall be mostly identified with integer types. Transforming a human readable message into an integer type is about encoding characters.

6.1 Symmetric Ciphers

The simplest symmetric cipher is the one know as Caesar's cipher. Take the alphabet and shift the position of its letters by a given number as pictured in 1. The ciphering map $\mathfrak{c}$ in this case is the one read up-down and sending a to f, b to g etc.

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
f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e

Figure 1: Alphabet shift by 6 for Caesar's cipher

Ciphering *hello* is thus given by

$$\mathfrak{c}(\text{hello}) = \text{mjqqt}$$

applying image of map character-wise. To get back original message you simply need to read previous table bottom-up. The map $\mathfrak{d}$ reads f goes to a, g goes to b, etc. Knowing $\mathfrak{d}$ or $\mathfrak{c}$ is about knowing the same table but reading it differently. Both sender and recipient of message have equivalent knowledge of cryptosystem. Having $\mathfrak{d}$ is enough to have $\mathfrak{c}$ and vice-versa. That's the reason why such a cryptosystem is said to be symmetric, knowledge of sender and recipient of the encryption and decryption data are equivalent.

Putting character encoding on the side, the Caesar cipher principle can be summed up in the following form : give yourself an integer $n > 1$ modulo which we'll be working, a Caesar cipher is then about the choice of an integer $k \in \mathbb{Z}$ corresponding to the shift. The map $\mathfrak{c} : \mathbb{Z}/n\mathbb{Z} \to \mathbb{Z}/n\mathbb{Z}$ is given by $\mathfrak{c}(x) = x+k$. Its inverse is simply given by the expression $\mathfrak{d}(x) = x-k$.

R The previous encryption system is wholly determined by the integer k. When an encryption is determined by such numerical data that numerical data is often called a key.

Question 6-2 How can you generalize the previous encryption system?

Question 6-3 Can you think of a way to discover (we say ***break***) a Caesar cipher, if you have enough alphabetical ciphered messages?

6.2 Asymmetric Ciphers

An asymmetric cryptosystem, in contrast with the previous case, is a cryptosystem where sender has as hard a time as any external party to know of the deciphering method. It involves the release of public data by the recipient allowing any party to send them a ciphered message, $\mathfrak{c}$ is then publicly accessible to all. The recipient is the only one to have the deciphering method $\mathfrak{d}$. The main point is to build up a system in such a way that $\mathfrak{d}$ is hard to figure out only having $\mathfrak{c}$.

6.2.1 RSA Cryptosystem

The ***RSA cryptosystem*** is based on the fact it takes time to factor an integer into its prime factors. In the case at hand this is going to be translated by : it is too time-consuming to extract a prime factor from a number which is a product of two huge prime numbers.

The RSA cryptosystem involves both public and private keys. The first is needed to build $\mathfrak{c}$ and is available to all, the second is only known by the recipients who make their public key available to all.

To build both data one needs two (distinct) huge ***prime*** numbers p and q, whose product will be denoted by $n = pq$, as well as an invertible element e in $\mathbb{Z}/\varphi(n)\mathbb{Z}$ whose inverse is written d.

Public key This is the data (n, e).

Private key This is the data d.

Before getting into the ciphering and deciphering methods let's take a minute to explain how to generate such keys. We've seen in 5 the multiplicative group of $\mathbb{Z}/n\mathbb{Z}$ is of cardinal

$$\varphi(n) = \varphi(p)\varphi(q) = (p-1)(q-1).$$

To find an invertible element $e \in \mathbb{Z}/\varphi(n)\mathbb{Z}$ we need to look for an element e which is coprime to $(p-1)(q-1)$.

Given public and private data for an RSA cryptosystem:

Ciphering Given a message $x \in \mathbb{Z}/n\mathbb{Z}$ the corresponding ciphered message is $x^e\ [n]$. The map $\mathfrak{c}$ is simply given by the expression $\mathfrak{c}(x) \equiv x^e\ [n]$.

Deciphering To decipher a message $y \in \mathbb{Z}/n\mathbb{Z}$ one looks at $y^d\ [n]$. The deciphering map is simply given by $\mathfrak{d}(y) \equiv y^d\ [n]$.

Notice both ciphering and deciphering methods are about taking exponents of an integer modulo n. Modular fast exponentiation is easy and quick to compute ; one of the requirements to be a usable cryptosystem.

Proposition 6.1 Previous data do indeed define a cryptosystem, i.e. $\mathfrak{d} \circ \mathfrak{c} = \mathrm{id}$.

Proof. Assume first we're given an invertible element $x \in (\mathbb{Z}/n\mathbb{Z})^\times$. We're willing to prove that, for any such x

$$\mathfrak{d}\big(\mathfrak{c}(x)\big) \equiv \mathfrak{d}\big(x^e\big) \equiv x^{ed} \equiv x\ [n].$$

By definition $ed \equiv 1\ [\varphi(n)]$. It means there is $k \in \mathbb{Z}$ such that

$$ed + k\varphi(n) = 1.$$

Following 4.4

$$x^{ed} \equiv x^{1-k\varphi(n)} \equiv x \times \left(x^{\varphi(n)}\right)^{-k} \equiv x\ [n].$$

Notice here that in case k is positive then we're looking at the negative power of 1 which has to be understood as the power of the inverse of 1 (i.e. 1).

Assume x is not invertible anymore. Using the Chinese remainder theorem we have that

$$(\mathbb{Z}/n\mathbb{Z})^\times = (\mathbb{Z}/p\mathbb{Z})^\times \times (\mathbb{Z}/q\mathbb{Z})^\times.$$

Since invertibles of $\mathbb{Z}/p\mathbb{Z}$ and $\mathbb{Z}/q\mathbb{Z}$ are all the non-zero elements, we get that non-invertible elements modulo n correspond to couples having the form $(0, x_2)$ for $x_2 \in \mathbb{Z}/q\mathbb{Z}$ or $(x_1, 0)$ for $x_1 \in \mathbb{Z}/p\mathbb{Z}$. If both entries are 0 then expected result is indeed satisfied. Let's assume that none of x_1 nor x_2 are 0. Using previous notation in the first case we get that

$$(0,x_2)^{ed} \equiv (0, x_2^{1-k\varphi(p)\varphi(q)}) \equiv \left(0, x_2 \times \left(x_2^{\varphi(q)}\right)^{-k\varphi(p)}\right) \equiv (0,x_2)\ [n]$$

which is what we expect. Symmetric case with x_1 is checked in a similar fashion. □

6.2.2 ElGamal Cryptosystem

The ElGamal Cryptosystem is based on the difficulty of solving equations of the type $a^k \equiv b\ [n]$ for a given fixed a, b and n, k being the variable of the equation. This is called the discrete logarithm problem[7]. It is a little more challenging than the RSA to write down.

ElGamal cryptosystem lives within publicly available data (p,g):

- p is a prime number modulo which the logarithm problem is difficult to solve.
- g is an element modulo p having high enough order[8].

Such data can be generated by the recipient or any other trusted party. Having such public data available the recipient chooses a private key $a \in \mathbb{Z}$ and computes and makes available their public key $A = g^a\ [p]$. Thus having public data (p,g)

Private key (Nearly) Any (big enough) chosen integer a.

Public key The integer $A = g^a$ modulo p.

Given a prime number p for which the logarithm problem is difficult enough, finding g is first about testing for numbers having high enough orders. The order of an element g is a divisor of $p-1$. For example if we take a prime number $p = 2q+1$ where q is still prime, then the order of g is either 1, 2, q or $2q$. Thus if g^2 is not 1 modulo p, then g must be of order q or $2q$ which can be high enough.

Given public data and an ElGamal public key:

Ciphering Given a message $x \in \mathbb{Z}/p\mathbb{Z}$, sender generates an ephemeral[9] random integer k. The ciphered message is the couple (c_1,c_2) where $c_1 = g^k$ and $c_2 = xA^k$. The map $\mathfrak{c}$ is then defined by the expression $\mathfrak{c}(x) = (g^k, xA^k)$ for a given ephemeral key k.

Deciphering To decipher a message $(y_1,y_2) \in \mathbb{Z}/p\mathbb{Z}$ recipient computes $(y_1^a)^{-1} y_2$. The map $\mathfrak{d}$ is defined by $\mathfrak{d}(y_1,y_2) = (y_1^a)^{-1} y_2$.

Proposition 6.2 Previous data do indeed define a cryptosystem, i.e. $\mathfrak{d} \circ \mathfrak{c} = \mathrm{id}$.

Proof. Using previous notation our aim is to prove that

$$(c_1^a)^{-1} c_2 = \left(g^{ak}\right)^{-1} xA^k = x$$

The first equality is nothing but the definition. For the second one, by construction

$$g^a \equiv A\ [p]$$

thus

$$\left(g^{ak}\right)^{-1} A^k x \equiv x\ [p]$$

and that is our expected result. □

[7] You're invited to ask your maths teachers for the reason why such question is related to logarithms.
[8] Recommended to be prime.
[9] Only to be used once!