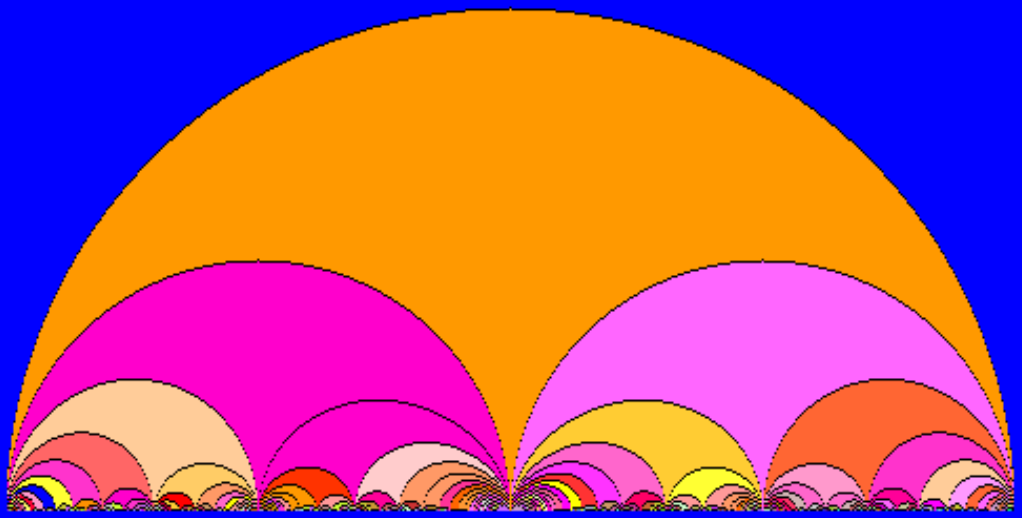

Yet Another Introductory Number Theory Textbook

(Cryptology Emphasis Version)

Jonathan A. Poritz

after Wissam Raji

Department of Mathematics and Physics
Colorado State University, Pueblo
2200 Bonforte Blvd.
Pueblo, CO 81001, USA
E-mail: jonathan.poritz@gmail.com
Web: www.poritz.net/jonathan

07 May 2014 11:04MDT

Preface

This is a first draft of a free (as in speech, not as in beer, [**Sta02**]) (although it is free as in beer as well) undergraduate number theory textbook. It was used for Math 319 at Colorado State University –Pueblo in the spring semester of 2014. Thanks are hereby offered to the students in that class – Megan Bissell, Tennille Candelaria, Ariana Carlyle, Michael Degraw, Daniel Fisher, Aaron Griffin, Lindsay Harder, Graham Harper, Helen Huang, Daniel Nichols, and Arika Waldrep – who offered many useful suggestions and found numerous typos. I am also grateful to the students in my Math 242 Introduction to Mathematical Programming class in that same spring semester of 2014 – Stephen Ciruli, Jamen Cox, Graham Harper, Joel Kienitz, Matthew Klamm, Christopher Martin, Corey Sullinger, James Todd, and Shelby Whalen – whose various programming projects produced code that I adapted to make some of the figures and examples in the text.

The author gratefully acknowledges the work **An Introductory Course in Elementary Number Theory** by Wissam Raji [see `www.saylor.org/books/`] from which this was initially adapted. Raji's text was released under the Creative Commons **CC BY 3.0** license, see `creativecommons.org/licenses/by/3.0`.

This version: 07 May 2014 11:04MDT. Note this text will be frequently updated and improved as the author has time, particularly during and immediately after semesters in which it is being used in a class. Therefore please check back often to the website, which is `www.poritz.net/jonathan/share/yaintt`.

This work is dedicated to my insanely hardworking colleagues at Colorado State University – Pueblo whose dedication to their students, their scholarship, and their communities is an inspiration. While I was working on the first version of this book, those colleagues stood up to some of the most benighted, ignorant administrative nonsense I have seen in the more than thirty years I have been involved in higher education. As MLK said, "The arc of the moral universe is long, but it bends towards justice." – It is selfless, intelligent, hard work like yours that is doing the bending.

Jonathan A. Poritz, 7 May 2014, Pueblo, CO, USA

Release Notes

This version of *YAINTT* has a particular emphasis on connections to cryptology. The cryptologic material appears in Chapter 4 and §§ 5.5 and 5.6, arising naturally (I hope) out of the ambient number theory. The main cryptologic applications – being the RSA cryptosystem, Diffie-Hellman key exchange, and the ElGamal cryptosystem – come out so naturally from considerations of Euler's Theorem, primitive roots, and indices that it renders quite ironic G.H. Hardy's assertion [**Har05**] of the purity and eternal inapplicability of number theory.

Note, however, that once we broach the subject of these cryptologic algorithms, we take the time to make careful definitions for many cryptological concepts and to develop some related ideas of cryptology which have much more tenuous connections to the topic of number theory. This material therefore has something of a different flavor from the rest of the text – as is true of all scholarly work in cryptology (indeed, perhaps in all of computer science), which is clearly a discipline with a different culture from that of "pure" mathematics. Obviously, these sections could be skipped by an uninterested reader, or remixed away by an instructor for her own particular class approach.

Caution: In good Bourbaki[1] style, where this symbol appears in the text below, it indicates a place where the reasoning is intricate and difficult to follow, or calls attention to a common misinterpretation of some point.

This version, in PDF form, can be found at

`http://www.poritz.net/jonathan/share/yaintt.pdf`

while all the files to create custom versions can be found at

`http://www.poritz.net/jonathan/share/yaintt/`

– have fun with it, that's the point of the Creative Commons!

[1] A fictional mathematician and author of many (non-fictional – they really exist) fine mathematics texts, such as [**Bou04**]

Contents

CHAPTER 1

Well-Ordering and Division

1.1. The Well-Ordering Principle and Mathematical Induction

In this chapter, we present three basic tools that will often be used in proving properties of the integers. We start with a very important property of integers called the well-ordering principle. We then state what is known as the pigeonhole principle, and then we proceed to present an important method called mathematical induction.

1.1.1. The Well-Ordering Principle.

DEFINITION 1.1.1. Given a set S of numbers (of any kind), we say that $\ell \in S$ is a **least element of** S if $\forall x \in S$, either $x = \ell$ or $\ell < x$.

THE WELL-ORDERING PRINCIPLE. Every non-empty set of natural numbers has a least element.

This principle is often taken as an axiom.

1.1.2. The Pigeonhole Principle.

THEOREM 1.1.2. ***The Pigeonhole Principle:*** *Let $s, k \in \mathbb{N}$ satisfy $s > k$. If s objects are placed in k boxes, then at least one box contains more than one object.*

PROOF. Suppose that none of the boxes contains more than one object. Then there are at most k objects. This leads to a contradiction with the fact that there are s objects for $s > k$. □

1.1.3. The Principle of Mathematical Induction. We now present a valuable tool for proving results about integers. This tool is the principle of mathematical induction.

THEOREM 1.1.3. ***The First Principle of Mathematical Induction:*** *Let $S \subset \mathbb{N}$ be a set satisfying the following two properties:*

(1) $1 \in S$*; and*

(2) $\forall k \in \mathbb{N},\ k \in S \Rightarrow k + 1 \in S$.

Then $S = \mathbb{N}$.

More generally, if $\mathcal{P}(n)$ is a property of natural numbers which may or may not be true for any particular $n \in \mathbb{N}$, satisfying

(1) $\mathcal{P}(1)$ *is true; and*

(2) $\forall k \in \mathbb{N},\ \mathcal{P}(k) \Rightarrow \mathcal{P}(k+1)$

then $\forall n \in \mathbb{N}, \mathcal{P}(n)$ *is true.*

PROOF. We use the well-ordering principle to prove this first principle of mathematical induction.

Let S be the set from the first part of the theorem and let T be the set of natural numbers not in S. We will use a proof by contradiction, so assume T is non-empty.

Then, by the well-ordering principle, T contains a least element ℓ.

Note that $1 \in S$, so $1 \notin T$ and thus $\ell > 1$. Therefore $\ell - 1$ is a natural number. Since ℓ is the least element of T, $\ell - 1$ is not in T, it is therefore in S.

But by the defining properties of S, since $\ell - 1 \in S$, $\ell = \ell - 1 + 1 \in S$, which contradicts the fact that ℓ is a least element of T, so in T, so not in S.

This contradiction implies that the assumption that T is non-empty is false, hence $S = \mathbb{N}$.

For the second part of the theorem, let $S = \{n \in \mathbb{N} \mid \mathcal{P}(n) \text{ is true}\}$ and apply the first part. □

EXAMPLE 1.1.4. We use mathematical induction to show that $\forall n \in \mathbb{N}$

$$\sum_{j=1}^{n} j = \frac{n(n+1)}{2}. \tag{1.1.1}$$

First note that

$$\sum_{j=1}^{1} j = 1 = \frac{1 \cdot 2}{2}$$

and thus the the statement is true for $n = 1$. For the remaining inductive step, suppose that the formula holds for some particular $n \in \mathbb{N}$, that is $\sum_{j=1}^{n} j = \frac{n(n+1)}{2}$. We show that

$$\sum_{j=1}^{n+1} j = \frac{(n+1)(n+2)}{2}.$$

to complete the proof by induction. Indeed

$$\sum_{j=1}^{n+1} j = \sum_{j=1}^{n} j + (n+1) = \frac{n(n+1)}{2} + (n+1) = \frac{(n+1)(n+2)}{2},$$

and the result follows.

EXAMPLE 1.1.5. Now we use mathematical induction to prove that $n! \leq n^n\ \forall n \in \mathbb{N}$.

Note that $1! = 1 \leq 1^1 = 1$. We now present the inductive step. Suppose that

$$n! \leq n^n$$

for some $n \in \mathbb{N}$, we prove that $(n+1)! \leq (n+1)^{n+1}$. Note that

$$(n+1)! = (n+1)n! \leq (n+1).n^n < (n+1)(n+1)^n = (n+1)^{n+1}.$$

This completes the proof.

THEOREM 1.1.6. ***The Second Principle of Mathematical Induction:*** *Let $S \subset \mathbb{N}$ be a set satisfying the following two properties:*

(1) $1 \in S$*; and*

(2) $\forall k \in \mathbb{N},\ 1, \ldots, k \in S \Rightarrow k+1 \in S$.

Then $S = \mathbb{N}$.

More generally, if $\mathcal{P}(n)$ is a property of natural numbers which may or may not be true for any particular $n \in \mathbb{N}$, satisfying

(1) $\mathcal{P}(1)$ *is true; and*

(2) $\forall k \in \mathbb{N}$, *if $\mathcal{P}(1), \ldots, \mathcal{P}(k)$ are all true, then $\mathcal{P}(k+1)$ is also true*

then $\forall n \in \mathbb{N}, \mathcal{P}(n)$ is true.

PROOF. To prove the second principle of induction, we use the first principle of induction.

Let S be a set of integers as in the first part of the theorem. For $n \in \mathbb{N}$, let $\mathcal{P}(n)$ be the mathematical property "$1, \ldots, n \in S$". Then we can apply the First Principle of Mathematical Induction to prove that $\forall n \in \mathbb{N}\ \mathcal{P}(n)$ is true, which means $S = \mathbb{N}$. *[Details left to the reader.]*

The second part of the theorem follows from the first in exactly the same way that the second part of the First Principle of Mathematical Induction followed from the first. □

Exercises for §1.1.

EXERCISE 1.1.1. Prove using mathematical induction that $n < 3^n$ for all positive integers n.

EXERCISE 1.1.2. Show that $\sum_{j=1}^{n} j^2 = \frac{n(n+1)(2n+1)}{6}$.

EXERCISE 1.1.3. Use mathematical induction to prove that $\sum_{j=1}^{n}(-1)^{j-1}j^2 = (-1)^{n-1}n(n+1)/2$.

EXERCISE 1.1.4. Use mathematical induction to prove that $\sum_{j=1}^{n} j^3 = [n(n+1)/2]^2$ for every positive integer n.

EXERCISE 1.1.5. Use mathematical induction to prove that $\sum_{j=1}^{n}(2j-1) = n^2$.

EXERCISE 1.1.6. Use mathematical induction to prove that $2^n < n!$ for $n \geq 4$.

EXERCISE 1.1.7. Use mathematical induction to prove that $n^2 < n!$ for $n \geq 4$.

1.2. Algebraic Operations with Integers

On $\mathbb{Z}$, the set of integers, there are two basic binary operations, namely **addition** (denoted by $+$) and **multiplication** (denoted by $\cdot$), which satisfy the following well known properties:

(1) **Commutativity of addition and multiplication**

$$\forall a, b \in \mathbb{Z}: \quad a + b = b + a$$
$$a \cdot b = b \cdot a$$

(2) **Associativity of addition and multiplication**

$$\forall a, b, c \in \mathbb{Z}: \quad (a + b) + c = a + (b + c)$$
$$(a \cdot b) \cdot c = a \cdot (b \cdot c)$$

(3) **Distributivity of multiplication over addition**

$$\forall a, b, c \in \mathbb{Z}: \quad a \cdot (b + c) = a \cdot b + a \cdot c.$$

In the set $\mathbb{Z}$ there are *identity elements* for the two operations $+$ and $\cdot$, and these are the elements 0 and 1 respectively, that satisfy the basic properties

$$\forall a \in \mathbb{Z}: \quad a + 0 = 0 + a = a$$
$$a \cdot 1 = 1 \cdot a = a\,.$$

The set $\mathbb{Z}$ allows **additive inverses** for its elements, in the sense that for every $a \in \mathbb{Z}$ there exists another integer in $\mathbb{Z}$, denoted by $-a$, such that

$$a + (-a) = 0. \tag{1.2.1}$$

While for multiplication, only the integer 1 has a **multiplicative inverse** in the sense that 1 is the only integer a such that there exists another integer, denoted by a^{-1} or by $1/a$, (namely 1 itself in this case) such that

$$a \cdot a^{-1} = 1. \tag{1.2.2}$$

From the operations of addition and multiplication one can define two other operations on $\mathbb{Z}$, namely **subtraction** (denoted by $-$) and **division** (denoted by $/$). Subtraction is a binary operation on $\mathbb{Z}$, *i.e.*, defined for any two integers in $\mathbb{Z}$, while division is not a binary operation and thus is defined only for some specific pairs of integers in $\mathbb{Z}$. Subtraction and division are defined as follows:

(1) $\forall a, b \in \mathbb{Z}$, $a - b$ is defined to be $a + (-b)$
(2) Given $a, b \in \mathbb{Z}$, where $b \neq 0$, if $\exists c \in \mathbb{Z}$ such that $a = b \cdot c$ then a/b is defined to be c.

1.3. Divisibility and the Division Algorithm

We now discuss the concept of divisibility and its properties.

1.3.1. Integer Divisibility.

DEFINITION 1.3.1. If a and b are integers such that $a \neq 0$, then we say a **divides** b and write $a \mid b$ if there exists an integer k such that $b = ka$. That is, given $a, b \in \mathbb{Z}$ such that $a \neq 0$, we write $a \mid b$ if $\exists k \in \mathbb{Z}$. s.t. $b = ka$.

If a divides b, we also say a is a **factor** [or **divisor**] of b, and b is a **multiple** of a. If a does not divide b, we write $a \nmid b$.

EXAMPLE 1.3.2. For example, $2 \mid 4$ and $7 \mid 63$, while $5 \nmid 26$.

DEFINITION 1.3.3. Given $a \in \mathbb{Z}$, we say a is **even** if $2 \mid a$, *i.e.,* if $\exists k \in \mathbb{Z}$ s.t. $a = 2k$. In contrast, given $a \in \mathbb{Z}$, we say a is **odd** if $2 \nmid a$.

It is a consequence of the Division Algorithm, below, that if a is odd then $\exists k \in \mathbb{Z}$ s.t. $a = 2k + 1$.

PROPOSITION 1.3.4. *$\forall a \in \mathbb{Z}$ we have $a \mid 0$.*

PROPOSITION 1.3.5. *If $b \in \mathbb{Z}$ is such that $|b| < a$, and $b \neq 0$, then $a \nmid b$.*

PROPOSITION 1.3.6. *Given $a, b \in \mathbb{Z}$, $a \mid b \Leftrightarrow a \mid |b|$.*

THEOREM 1.3.7. *If a, b and c are integers such that $a \mid b$ and $b \mid c$, then $a \mid c$.*

PROOF. Since $a \mid b$ and $b \mid c$, we know $\exists k_1, k_2 \in \mathbb{Z}$ such that $b = k_1 a$ and $c = k_2 b$. Hence $c = k_1 k_2 a$ and so $a \mid c$. □

EXAMPLE 1.3.8. Since $6 \mid 18$ and $18 \mid 36$, then $6 \mid 36$.

The following theorem states that if an integer divides two other integers then it divides any linear combination of these integers.

THEOREM 1.3.9. *$\forall a, b, c, m, n \in \mathbb{Z}$, if $c \mid a$ and $c \mid b$ then $c \mid (ma + nb)$.*

PROOF. Since $c \mid a$ and $c \mid b$, $\exists k_1, k_2 \in \mathbb{Z}$ such that $a = k_1 c$ and $b = k_2 c$. Thus

$$ma + nb = mk_1 c + nk_2 c = c(mk_1 + nk_2),$$

and hence $c \mid (ma + nb)$. □

Theorem 1.3.9 can be generalized to any finite linear combination as follows. If

$$n \in \mathbb{N},\ a, b_1, \ldots, b_n \in \mathbb{Z} \text{ and } a \mid b_1, a \mid b_2, \ldots, a \mid b_n$$

then

$$a \mid \sum_{j=1}^{n} k_j b_j \tag{1.3.1}$$

$\forall k_1, \ldots, k_n \in \mathbb{Z}$. It would be a nice exercise to prove this generalization by induction.

1.3.2. The Division Algorithm.

THEOREM 1.3.10. ***The Division Algorithm*** *Given $a, b \in \mathbb{Z}$ such that $b > 0$, there exist unique $q, r \in \mathbb{Z}$ such that $a = qb + r$ and $0 \leq r < b$. This q is called the* **quotient** *and r the* **remainder when a is divided by b.**

PROOF. Consider the set $A = \{a - bk \geq 0 \mid k \in \mathbb{Z}\}$. Note that A is nonempty since for $k < a/b$, $a - bk > 0$. By the well-ordering principle, A has a least element $r = a - qb$ for some $q \in \mathbb{Z}$. Notice that $r \geq 0$ by construction. Now if $r \geq b$ then (since $b > 0$)

$$r > r - b = a - qb - b = a - (q + 1)b \geq 0.$$

This leads to a contradiction since r is assumed to be the least positive integer of the form $r = a - qb$. As a result we have $0 \leq r < b$.

We will show that q and r are unique. Suppose that $a = q_1 b + r_1$ and $a = q_2 b + r_2$ with $0 \leq r_1 < b$ and $0 \leq r_2 < b$. Then we have

$$a - a = q_1 b + r_1 - (q_2 b + r_2) = (q_1 - q_2)b + (r_1 - r_2) = 0.$$

As a result we have

$$(q_1 - q_2)b = r_2 - r_1.$$

Thus we get that

$$b \mid (r_2 - r_1).$$

And since $-\max(r_1, r_2) \leq |r_2 - r_1| \leq \max(r_1, r_2)$, and $b > \max(r_1, r_2)$, then $r_2 - r_1$ must be 0, i.e. $r_2 = r_1$. And since $bq_1 + r_1 = bq_2 + r_2$, we also get that $q_1 = q_2$. This proves uniqueness. □

EXAMPLE 1.3.11. If $a = 71$ and $b = 6$, then $71 = 6 \cdot 11 + 5$. Here $q = 11$ and $r = 5$.

Exercises for §1.3.

EXERCISE 1.3.1. Show that $5 \mid 25, 19 \mid 38$ and $2 \mid 98$.

EXERCISE 1.3.2. Use the division algorithm to find the quotient and the remainder when 76 is divided by 13.

EXERCISE 1.3.3. Use the division algorithm to find the quotient and the remainder when -100 is divided by 13.

EXERCISE 1.3.4. Show that if a, b, c and d are integers with a and c nonzero, such that $a \mid b$ and $c \mid d$, then $ac \mid bd$.

EXERCISE 1.3.5. Show that if a and b are positive integers and $a \mid b$, then $a \leq b$.

EXERCISE 1.3.6. Prove that the sum of two even integers is even, the sum of two odd integers is even and the sum of an even integer and an odd integer is odd.

EXERCISE 1.3.7. Show that the product of two even integers is even, the product of two odd integers is odd and the product of an even integer and an odd integer is even.

EXERCISE 1.3.8. Show that if m is an integer then 3 divides $m^3 - m$.

EXERCISE 1.3.9. Show that the square of every odd integer is of the form $8m + 1$.

EXERCISE 1.3.10. Show that the square of any integer is of the form $3m$ or $3m + 1$ but not of the form $3m + 2$.

EXERCISE 1.3.11. Show that if $ac \mid bc$, then $a \mid b$.

EXERCISE 1.3.12. Show that if $a \mid b$ and $b \mid a$ then $a = \pm b$.

1.4. Representations of Integers in Different Bases

In this section, we show how any positive integer can be written in terms of any positive base integer expansion in a unique way. Normally we use decimal notation to represent integers, we will show how to convert an integer from decimal notation into any other positive base integer notation and vise versa. Using the decimal notation in daily life is more traditional probably only because we have ten fingers. ("What about our toes?" you cry. I don't know. And apparently the Babylonians had 30 fingers on each hand, or 15 on each hand and each foot, since they used base 60.)

Notation An integer a written in base b expansion is denoted by $(a)_b$.

THEOREM 1.4.1. *Let $b \in \mathbb{Z}$ satisfy $b > 1$. Then $\forall m \in \mathbb{N}$, $\exists l \in \mathbb{N}$ and $\exists a_1, \ldots, a_l \in \mathbb{Z}$ such that*

$$m = a_l b^l + a_{l-1} b^{l-1} + \cdots + a_1 b + a_0,$$
$$0 \leq a_j < b \text{ for } j = 0, 1, \ldots, l, \text{ and}$$
$$a_l \neq 0.$$

PROOF. Fix an $m \in \mathbb{N}$. We start by dividing m by b and we get

$$m = q_0 b + a_0, \quad 0 \leq a_0 < b.$$

If $q_0 \neq 0$ then we continue to divide q_0 by b and we get

$$q_0 = q_1 b + a_1, \quad 0 \leq a_1 < b.$$

We continue this process and hence we get

$$q_1 = q_2 b + a_2, \quad 0 \leq a_2 < b,$$
$$\vdots$$
$$q_{l-2} = q_{l-1} b + a_{l-1}, \quad 0 \leq a_{l-1} < b,$$
$$q_{l-1} = 0 \cdot b + a_l, \quad 0 \leq a_l < b.$$

Note that the sequence $q_0, q_1, \ldots$ is a decreasing sequence of non-negative integers with a last term q_l that must be 0.

Now substituting the equation $q_0 = q_1 b + a_1$ in $m = q_0 b + a_0$, we get

$$m = (q_1 b + a_1) b + a_0 = q_1 b^2 + a_1 b + a_0,$$

Successively substituting the equations in m, we get

$$\begin{aligned} m &= q_2 b^3 + a_2 b^2 + a_1 b + a_0, \\ &\vdots \\ &= q_{l-1} b^l + a_{l-1} b^{l-1} + \cdots + a_1 b + a_0, \\ &= a_l b^l + a_{l-1} b^{l-1} + \cdots + a_1 b + a_0. \end{aligned}$$

What remains to prove is that the representation is unique. Suppose now that

$$m = a_l b^l + a_{l-1} b^{l-1} + \cdots + a_1 b + a_0 = c_l b^l + c_{l-1} b^{l-1} + \cdots + c_1 b + c_0$$

where if the number of terms is different in one expansion, we add zero coefficients to make the number of terms agree. Subtracting the two expansions, we get

$$(a_l - c_l) b^l + (a_{l-1} - c_{l-1}) b^{l-1} + \cdots + (a_1 - c_1) b + (a_0 - c_0) = 0.$$

If the two expansions are different, then there exists $0 \leq j \leq l$ such that $c_j \neq a_j$. As a result, we get

$$b^j ((a_l - c_l) b^{l-j} + \cdots + (a_{j+1} - c_{j+1}) b + (a_j - c_j)) = 0$$

and since $b \neq 0$, we get

$$(a_l - c_l) b^{l-j} + \cdots + (a_{j+1} - c_{j+1}) b + (a_j - c_j) = 0.$$

We now get

$$a_j - c_j = (a_l - c_l) b^{l-j} + \cdots + (a_{j+1} - c_{j+1}) b,$$

and as a result, $b \mid (a_j - c_j)$. Since $0 \leq a_j < b$ and $0 \leq c_j < b$, we get that $a_j = c_j$. This is a contradiction and hence the expansion is unique. □

DEFINITION 1.4.2. Given $b \in \mathbb{Z}$ satisfying $b > 1$. For $m \in \mathbb{N}$, let $\ell \in \mathbb{N}$ and $a_1, \ldots, a_\ell \in \mathbb{Z}$ be as in the above theorem (1.4.1). Then the **base** b **expression for** m is the sequences of digits $m_b = a_\ell \ldots a_1$. If $b \geq 10$, we often use some other single symbols to represent the possible values from 10 to $b - 1$ of the a_i's. For example,

$$10 \leftrightsquigarrow A$$

$$11 \leftrightsquigarrow B$$

$$12 \leftrightsquigarrow C$$

etc.

Base 2 representation of integers is called **binary representation**. Binary representation is useful for computers: the coefficients $a_0, \ldots, a_l$ of a binary representation all satisfy $0 \leq aj < 2$, hence they are 0 or 1. Thus to represent an integer on l wires, one can have

each wire either have voltage (1) or not (0). (In fact, the word *bit* is a contraction of *binary digit*.)

Computer programmers also frequently use base 8 and base 16, called **octal** and **hexadecimal** or **hex**, respectively. The Babylonians used base 60, called **sexagesimal**.

EXAMPLE 1.4.3. To find the expansion of 214 base 3: we do the following

$$\begin{aligned} 214 &= 3 \cdot 71 + 1 \\ 71 &= 3 \cdot 23 + 2 \\ 23 &= 3 \cdot 7 + 2 \\ 7 &= 3 \cdot 2 + 1 \\ 2 &= 3 \cdot 0 + 2 \end{aligned}$$

As a result, to obtain a base 3 expansion of 214, we take the remainders of divisions and we get that $(214)_{10} = (21221)_3$.

EXAMPLE 1.4.4. To find the base 10 expansion, *i.e.,* the decimal expansion, of $(364)_7$: We do the following: $4 \cdot 7^0 + 6 \cdot 7^1 + 3 \cdot 7^2 = 4 + 42 + 147 = 193$.

1.4.1. Exercises for §1.4.

EXERCISE 1.4.1. Convert $(7482)_{10}$ to base 6 notation.

EXERCISE 1.4.2. Convert $(98156)_{10}$ to base 8 notation.

EXERCISE 1.4.3. Convert $(101011101)_2$ to decimal notation.

EXERCISE 1.4.4. Convert $(AB6C7D)_{16}$ to decimal notation.

EXERCISE 1.4.5. Convert $(9A0B)_{16}$ to binary notation.

1.5. The Greatest Common Divisor

In this section we define the greatest common divisor (gcd) of two integers and discuss its properties. We also prove that the greatest common divisor of two integers is a linear combination of these integers.

Two integers a and b, not both 0, can have only finitely many divisors (see Exercise 1.3.5), and thus can have only finitely many divisors in common. In this section, we are interested in the greatest of these common divisors.

DEFINITION 1.5.1. Given $a, b \in \mathbb{Z}$, not both zero, the **greatest common divisor** is the largest integer that divides both a and b, and is written $\gcd(a, b)$ (or sometimes just (a, b)).

When it makes some formulæ simpler, we will write $\gcd(0, 0) = 0$.

EXAMPLE 1.5.2. The greatest common divisor of 24 and 18 is 6. In other words $\gcd(24, 18) = 6$.

DEFINITION 1.5.3. $a, b \in \mathbb{Z}$ are said to be **relatively prime** if $\gcd(a, b) = 1$.

EXAMPLE 1.5.4. The greatest common divisor of 9 and 16 is 1, thus they are relatively prime.

Note that every integer has positive and negative divisors. If a is a positive divisor of m, then $-a$ is also a divisor of m. Therefore by our definition of the greatest common divisor, we can see that $\gcd(a, b) = \gcd(|a|, |b|)$.

We can use the gcd of two integers to make relatively prime integers:

THEOREM 1.5.5. *If* $a, b \in \mathbb{Z}$ *have* $\gcd(a, b) = d$ *then* $\gcd(a/d, b/d) = 1$.

PROOF. Fix $a, b \in \mathbb{Z}$. We will show that a/d and b/d have no common positive divisors other than 1. Let $k \in \mathbb{N}$ be a divisor of both a/d and b/d, so $\exists m, n \in \mathbb{N}$ such that

$$a/d = km \quad \text{and} \quad b/d = kn$$

Thus we get that

$$a = kmd \quad \text{and} \quad b = knd.$$

Hence kd is a common divisor of both a and b. Also, $kd \geq d$. However, d is the greatest common divisor of a and b. As a result, we get that $k = 1$. □

The next theorem shows that the greatest common divisor of two integers does not change when we add a multiple of one of the two integers to the other.

THEOREM 1.5.6. *Let* $a, b, c \in \mathbb{Z}$. *Then* $\gcd(a, b) = \gcd(a + cb, b)$.

PROOF. We will show that every divisor of a and b is also a divisor of $a + cb$ and b and vise versa. Hence they have exactly the same divisors. So we get that the greatest common divisor of a and b will also be the greatest common divisor of $a + cb$ and b. Let k be a

common divisor of a and b. By Theorem 1.3.9, $k \mid (a + cb)$ and hence k is a divisor of $a + cb$. Now assume that l is a common divisor of $a + cb$ and b. Also by Theorem 1.3.9 we have,

$$l \mid ((a + cb) - cb) = a.$$

As a result, l is a common divisor of a and b and the result follows. □

EXAMPLE 1.5.7. Notice that $\gcd(4, 14) = \gcd(4, 14 - 3 \cdot 4) = \gcd(4, 2) = 2$.

We now present a theorem which proves that the greatest common divisor of two integers can be written as a linear combination of the two integers.

THEOREM 1.5.8. *Let $a, b \in \mathbb{Z}$ not both be zero. Then $\gcd(a, b)$ is the smallest natural number which is of the form $d = ma + nb$ for some $m, n \in \mathbb{Z}$.*

PROOF. Assume without loss of generality that $a, b \in \mathbb{N}$ are positive integers. Consider the set

$$S = \{d \in \mathbb{N} \mid d = ma + nb \text{ for some } m, n \in \mathbb{Z}\} .$$

S is non-empty since $a = 1 \cdot a + 0 \cdot b$ and $b = 0 \cdot a + 1 \cdot b$ are both in S. Let $d \in \mathbb{N}$ be the least element of S, whose existence is guaranteed by the well-ordering principle. Notice $d = ma + nb$ for some $m, n \in \mathbb{Z}$, since $d \in S$. We still must prove that d divides both a and b and that it is the greatest such common divisor.

By the division algorithm, $\exists q, r \in \mathbb{Z}$ such that

$$a = qd + r, \quad 0 \leq r < d.$$

Thus we have

$$r = a - qd = a - q(ma + nb) = (1 - qm)a - qnb.$$

We then have that r is a linear combination of a and b. Since $0 \leq r < d$ and d is the least positive integer which is a linear combination of a and b, we must have $r = 0$ and so $a = qd$. Hence $d \mid a$.

The same sort of argument will show that $d \mid b$.

Now notice that if there is a divisor c that divides both a and b. Then c divides any linear combination of a and b by Theorem 1.3.9. Hence $c \mid d$. This proves that any common divisor of a and b divides d. Hence $c \leq d$, and d is the greatest common divisor. □

There is a simple application of this which will be very useful in the future:

COROLLARY 1.5.9. *If $a, b \in \mathbb{Z}$ are relatively prime, then $\exists m, n \in \mathbb{Z}$ such that $ma + nb = 1$.*

DEFINITION 1.5.10. For some $n \in \mathbb{N}$, let $a_1, a_2, \ldots, a_n \in \mathbb{Z}$ not be all 0. The **greatest common divisor** of these integers is the largest integer that divides all of them, and is denoted $\gcd(a_1, \ldots, a_n)$.

DEFINITION 1.5.11. For some $n \in \mathbb{N}$, $a_1, a_2, \ldots, a_n \in \mathbb{Z}$ are said to be **mutually relatively prime** if $\gcd(a_1, a_2, \ldots, a_n) = 1$.

EXAMPLE 1.5.12. The integers $3, 6, 7$ are mutually relatively prime since $(3, 6, 7) = 1$ although $(3, 6) = 3$.

DEFINITION 1.5.13. For some $n \in \mathbb{N}$, $a_1, a_2, \ldots, a_n \in \mathbb{Z}$ are called **pairwise relatively prime** if $\forall i, j \in \mathbb{N}$ such that $i \leq n$, $j \leq n$, and $i \neq j$, we have $\gcd(a_i, a_j) = 1$.

EXAMPLE 1.5.14. The integers $3, 14, 25$ are pairwise relatively prime. Notice also that these integers are mutually relatively prime.

PROPOSITION 1.5.15. *For $n \in \mathbb{N}$ and $a_1, \ldots, a_n \in \mathbb{Z}$, if $a_1, a_2, \ldots, a_n$ are pairwise relatively prime then they are mutually relatively prime.*

Exercises for §1.5.

EXERCISE 1.5.1. Find the greatest common divisor of 15 and 35.

EXERCISE 1.5.2. Find the greatest common divisor of 100 and 104.

EXERCISE 1.5.3. Find the greatest common divisor of -30 and 95.

EXERCISE 1.5.4. Let $m \in \mathbb{N}$. Find the greatest common divisor of m and $m + 1$.

EXERCISE 1.5.5. Let $m \in \mathbb{N}$, find the greatest common divisor of m and $m + 2$.

EXERCISE 1.5.6. Show that if $m, n \in \mathbb{Z}$ have $\gcd(m, n) = 1$, then $\gcd(m+n, m-n) = 1$ or 2.

EXERCISE 1.5.7. Show that if $m \in \mathbb{N}$, then $3m + 2$ and $5m + 3$ are relatively prime.

EXERCISE 1.5.8. Show that if $a, b \in \mathbb{Z}$ are relatively prime, then $\gcd(a+2b, 2a+b) = 1$ or 3.

EXERCISE 1.5.9. Show that if $a_1, a_2, \ldots, a_n \in \mathbb{Z}$ are not all 0 and $c \in \mathbb{N}$, then $\gcd(ca_1, ca_2, \ldots, ca_n) = c \cdot \gcd(a_1, a_2, \ldots, a_n)$.

1.6. The Euclidean Algorithm

In this section we describe a systematic method that determines the greatest common divisor of two integers, due to Euclid and thus called the Euclidean algorithm.

LEMMA 1.6.1. *If $a, b, q, r \in \mathbb{Z}$ and $a = qb + r$, then $\gcd(a, b) = \gcd(r, b)$.*

PROOF. Note that by theorem 8, we have $\gcd(bq + r, b) = \gcd(b, r)$. □

Now to the Euclidean algorithm in its general form, which basically states that the greatest common divisor of two integers is the last non zero remainder of successive divisions.

THEOREM 1.6.2. *Let $a, b \in \mathbb{N}$ and assume $a \geq b$. Define $r_0 = a$, $r_1 = b$, $s_0 = 1$, $s_1 = 0$, $t_0 = 0$, and $t_1 = 1$. Then apply the division algorithm successively to obtain quotients and remainders $q_j, r_j \in \mathbb{N}$ satisfying $r_j = r_{j+1}q_{j+1} + r_{j+2}$ and $0 \leq r_{j+2} < r_{j+1}$ for all $j = 0, 1, \ldots, n-2$ where n is defined so that $r_{n+1} = 0$. Along the way, also define $s_{j+1} = s_{j-1} - q_{j+1}s_j$ and $t_{j+1} = t_{j-1} - q_{j+1}t_j$. Then $\gcd(a, b) = r_n = s_{n+1}a + t_{n+1}b$.*

PROOF. By applying the division algorithm, we see that

$$\begin{aligned}
r_0 &= q_1 r_1 + r_2 \quad && 0 \leq r_2 < r_1, \\
r_1 &= q_2 r_2 + r_3 \quad && 0 \leq r_3 < r_2, \\
&\vdots \\
r_{n-2} &= q_{n-1} r_{n-1} + r_n \quad && 0 \leq r_n < r_{n-1}, \\
r_{n-1} &= q_n r_n.
\end{aligned}$$

Notice that, we will have a remainder of 0 eventually since all the remainders are integers and every remainder in the next step is less than the remainder in the previous one. By Lemma 1.6.1, we see that

$$\gcd(a, b) = \gcd(b, r_2) = \gcd(r_2, r_3) = \cdots = \gcd(r_n, 0) = r_n.$$

□

Note: The full version of this theorem, with the s_j's and t_j, is called the **extended Euclidean Algorithm**, while a simpler version without those coefficients is know as **Euclidean Algorithm**.

The attentive reader will have seen that We did not actually prove that the s_j's and t_j's can be used, as claimed, to write the $\gcd$ as a linear combination of a and b. This proof is left as an exercise, below.

EXAMPLE 1.6.3. We will find the greatest common divisor of 4147 and 10672: Note that

$$\begin{aligned} 10672 &= 4147 \cdot 2 + 2378, \\ 4147 &= 2378 \cdot 1 + 1769, \\ 2378 &= 1769 \cdot 1 + 609, \\ 1769 &= 609 \cdot 2 + 551, \\ 609 &= 551 \cdot 1 + 58, \\ 551 &= 58 \cdot 9 + 29, \\ 58 &= 29 \cdot 2, \end{aligned}$$

Hence $\gcd(4147, 10672) = 29$.

Exercises for §1.6.

EXERCISE 1.6.1. Use the Euclidean algorithm to find the greatest common divisor of 412 and 32 and express it in terms of the two integers.

EXERCISE 1.6.2. Use the Euclidean algorithm to find the greatest common divisor of 780 and 150 and express it in terms of the two integers.

EXERCISE 1.6.3. Find the greatest common divisor of $70, 98, 108$.

EXERCISE 1.6.4. Let $a, b \in \mathbb{N}$ be even. Prove that $\gcd(a, b) = 2\gcd(a/2, b/2)$.

EXERCISE 1.6.5. Show that if $a \in \mathbb{N}$ is even and $b \in \mathbb{N}$ is odd, then $\gcd(a, b) = \gcd(a/2, b)$.

EXERCISE 1.6.6. Prove the *extended* part of the Extended Euclidean Algorithm.

CHAPTER 2

Congruences

A congruence is nothing more than a statement about divisibility. The theory of congruences was introduced by Carl Friedrich Gauss, in his monumental *Disquisitiones Arithmeticae* (published in 1801, when he was 24;a translation is [**Gau86**]).

We start by introducing congruences and their properties. We then present solutions to linear congruences which will serve as an introduction to the Chinese Remainder Theorem that follows.

2.1. Introduction to Congruences

As we mentioned in the introduction, the theory of congruences was developed by Gauss at the beginning of the nineteenth century.

DEFINITION 2.1.1. Given $a, b \in \mathbb{Z}$ and $n \in \mathbb{N}$, we say that a **is congruent to** b **modulo** n if $n \mid (a-b)$, *i.e.,* if $\exists k \in \mathbb{Z}$ such that $a = b + kn$. If a is congruent to b modulo n, we write $a \equiv b \pmod{n}$.

EXAMPLE 2.1.2. $19 \equiv 5 \pmod{7}$. Similarly $2k+1 \equiv 1 \pmod{2}$ which means every odd number is congruent to 1 modulo 2.

Congruence is much like equality in many ways. For example:

THEOREM 2.1.3. *Given* $a, b, c, d \in \mathbb{Z}$ *and* $n \in \mathbb{N}$. *Then*

(1) *If* $a \equiv b \pmod{n}$, *then* $b \equiv a \pmod{n}$.
(2) *If* $a \equiv b \pmod{n}$ *and* $b \equiv c \pmod{n}$, *then* $a \equiv c \pmod{m}$.
(3) *If* $a \equiv b \pmod{n}$, *then* $a + c \equiv b + c \pmod{n}$.
(4) *If* $a \equiv b \pmod{n}$, *then* $a - c \equiv b - c \pmod{n}$.
(5) *If* $a \equiv b \pmod{n}$, *then* $ac \equiv bc \pmod{n}$.
(6) *If* $c > 0$ *and* $a \equiv b \pmod{n}$, *then* $ac \equiv bc \pmod{nc}$.
(7) *If* $a \equiv b \pmod{n}$ *and* $c \equiv d \pmod{n}$ *then* $a + c \equiv b + d \pmod{n}$.
(8) *If* $a \equiv b \pmod{n}$ *and* $c \equiv d \pmod{n}$ *then* $a - c \equiv b - d \pmod{n}$.
(9) *If* $a \equiv b \pmod{n}$ *and* $c \equiv d \pmod{n}$ *then* $ac \equiv bd \pmod{n}$.

PROOF.

(1) If $a \equiv b \pmod{n}$, then $n \mid (a-b)$. Thus $\exists k \in \mathbb{Z}$ such that $a - b = kn$. This implies $b - a = (-k)n$ and thus $n \mid (b-a)$. Consequently $b \equiv a \pmod{n}$.

(2) Since $a \equiv b \pmod{n}$ and $b \equiv c \pmod{n}$, $n \mid (a-b)$ and $n \mid (b-c)$. As a result, there $\exists k, l \in \mathbb{Z}$ such that $a = b + kn$ and $b = c + ln$, which imply that $a = c + (k+l)n$. In other words, $a = c \pmod{n}$.

(3) Since $a \equiv b \pmod{n}$, $n \mid (a-b)$. So if we add and subtract c we get

$$n \mid ((a+c) - (b+c))$$

which means that

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

(4) Since $a \equiv b \pmod{n}$, $n \mid (a-b)$ so we can subtract and add c to get

$$n \mid ((a-c) - (b-c))$$

so

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

(5) If $a \equiv b \pmod{n}$, $n \mid (a-b)$. Thus there $\exists k \in \mathbb{Z}$ such that $a - b = kn$ and as a result $ac - bc = (kc)n$. Therefore

$$n \mid (ac - bc)$$

and hence

$$ac \equiv bc \pmod{n}.$$

(6) If $a \equiv b \pmod{n}$, $n \mid (a-b)$. Thus there $\exists k \in \mathbb{Z}$ such that $a - b = kn$ and as a result

$$ac - bc = (k)cn.$$

Thus

$$nc \mid (ac - bc)$$

and hence

$$ac \equiv bc \pmod{nc}.$$

(7) Since $a \equiv b \pmod{n}$ and $c \equiv d \pmod{n}$, $n \mid (a-b)$ and $n \mid (c-d)$. As a result, there $\exists k, l \in \mathbb{Z}$ such that $a - b = kn$ and $c - d = ln$. Note that

$$(a-b) + (c-d) = (a+c) - (b+d) = (k+l)n.$$

As a result,

$$n \mid ((a+c) - (b+d)),$$

hence

$$a + c \equiv b + d \pmod{n}.$$

(8) If $a = b + kn$ and $c = d + ln$ for $k, l \in \mathbb{Z}$, we have

$$(a - b) - (c - d) = (a - c) - (b - d) = (k - l)n.$$

As a result,

$$n \mid ((a - c) - (b - d)),$$

hence

$$a - c \equiv b - d \pmod{n}.$$

(9) $\exists k, l \in \mathbb{Z}$ such that such that $a - b = kn$ and $c - d = ln$ and thus $ca - cb = (ck)n$ and $bc - bd = (bl)n$. Note that

$$(ca - cb) + (bc - bd) = ac - bd = (kc - lb)n.$$

As a result,

$$n \mid (ac - bd),$$

hence

$$ac \equiv bd \pmod{n}.$$

□

Here is a technical result which will be useful later:

THEOREM 2.1.4. *Given $a, b, c \in \mathbb{Z}$, if $a \mid c$, $b \mid c$, and a and b are relatively prime, then $ab \mid c$.*

PROOF. By Corollary 1.5.9, we know $\exists m, n \in \mathbb{Z}$ such that $ma + nb = 1$. Also, because of the divisibility hypotheses, we also know $\exists p, q \in \mathbb{Z}$ such that $c = pa$ and $c = qb$. Compute:

$$c = c \cdot 1 = c(ma + nb) = mca + ncb = mqba + npab = (mq + np)ab\ .$$

But this means $ab \mid c$, as desired. □

EXAMPLES 2.1.5.

(1) $14 \equiv 8 \pmod{6}$ *so* $8 \equiv 14 \pmod{6}$.
(2) *Because* $22 \equiv 10 \pmod{6}$ *and* $10 \equiv 4 \pmod{6}$, *it is also true that* $22 \equiv 4 \pmod{6}$.
(3) $50 \equiv 20 \pmod{15}$ *so* $50 + 5 = 55 \equiv 20 + 5 = 25 \pmod{15}$.
(4) $50 \equiv 20 \pmod{15}$ *so* $50 - 5 = 45 \equiv 20 - 5 = 15 \pmod{15}$.
(5) $19 \equiv 16 \pmod{3}$ *so* $2(19) = 38 \equiv 2(16) = 32 \pmod{3}$.
(6) $19 \equiv 16 \pmod{3}$ *so* $2(19) = 38 \equiv 2(16) = 32 \pmod{2 \cdot 3}$ *or* $38 \equiv 2(16) = 32 \pmod{6}$.
(7) *Because* $19 \equiv 3 \pmod{8}$ *and* $17 \equiv 9 \pmod{8}$, *we have* $19 + 17 = 36 \equiv 3 + 9 = 12 \pmod{8}$.

(8) *Because* $19 \equiv 3 \pmod 8$ *and* $17 \equiv 9 \pmod 8$*, we have* $19 - 17 = 2 \equiv 3 - 9 = -6 \pmod 8$.
(9) *Because* $19 \equiv 3 \pmod 8$ *and* $17 \equiv 9 \pmod 8$*, we have* $19(17) = 323 \equiv 3(9) = 27 \pmod 8$.

Here is a result which at first seems very simple, but turns out to be immensely useful – so useful it has a name.

LEMMA 2.1.6. **Euclid's Lemma:** *Given* $x, y, z \in \mathbb{Z}$*, if* $x \mid yz$ *and* $\gcd(x, y) = 1$ *then* $x \mid z$.

PROOF. From Corollary 1.5.9, we know $\exists m, n \in \mathbb{Z}$ such that $mx + ny = 1$. Multiplying by z, we get $mxz + nyz = z$. But we've assumed that $x \mid yz$, so $x \mid nyz$, and certainly $x \mid mxz$, so $x \mid mxz + nyz$, *i.e.,* $x \mid z$. □

We now present a theorem that will show one difference between equations and congruences: in equations, if we divide both sides of the equation by a non-zero number, equality holds. However, in congruences, this is not necessarily true. In other words, dividing both sides of a congruence by the same integer does not necessarily preserve the congruence.

THEOREM 2.1.7.

(1) *Given* $a, b, c \in \mathbb{Z}$ *and* $n \in \mathbb{N}$*, define* $d = \gcd(a, n) \in \mathbb{N}$*. If* $ab \equiv ac \pmod n$ *then* $b \equiv c \pmod{n/d}$.
(2) *In particular, if* $\gcd(a, n) = 1$ *then*

$$b = c \pmod n \qquad \Leftrightarrow \qquad ab \equiv ac \pmod n.$$

PROOF. For Part 1, if $ab \equiv ac \pmod n$, then

$$n \mid (ab - ac) = a(b - c).$$

Hence $\exists k \in \mathbb{Z}$ such that $a(b - c) = kn$. Dividing both sides by d, we get $(a/d)(b - c) = k(n/d)$ or $(n/d) \mid (a/d)(b - c)$. Now, by Theorem 1.5.5 $\gcd(a/d, n/d) = 1$ so Euclid's Lemma 2.1.6 tells us that $(n/d) \mid (b - c)$. Hence $b \equiv c \pmod{n/d}$.

For Part 2, the direction $\Rightarrow$ is part 5 of Theorem 2.1.3, while $\Leftarrow$ is a special case of Part 1. □

EXAMPLE 2.1.8. $38 \equiv 10 \pmod 7$. Since $\gcd(2, 7) = 1$, we have $19 \equiv 5 \pmod 7$.

One last technical result is worth stating clearly at this point:

THEOREM 2.1.9. *Given* $n, d \in \mathbb{N}$ *such that* $d \mid n$*, there are exactly* d *values* $x \in \mathbb{Z}$*, up to congruence modulo* n*, satisfying* $x \equiv 0 \pmod{n/d}$.

PROOF. Let $x_j = j(n/d)$ for $j = 0, \ldots, (d-1)$. Certainly each of these d values x_j is a multiple of n/d and so solves $x \equiv 0 \pmod{n/d}$. All we must show, then, is that *every* solution x of $x \equiv 0 \pmod{n/d}$ is congruent, modulo n, to one of these x_j.

Let x be such a solution, so $\exists k \in \mathbb{Z}$ such that $x = k(n/d)$. Use the Division Algorithm for x divided by n, getting $x = qn + r$ for some $q, r \in \mathbb{Z}$ with $0 \leq r < n$. But

$$r = x - qn = k(n/d) - qd(n/d) = (k - qd)(n/d)$$

so r is a multiple of (n/d) which lies in the range $[0, n)$. The only such multiples are the $x_0, \ldots, x_{(d-1)}$ defined above; say $r = x_j$. Then $x = qn + r = qn + x_j \equiv x_j \pmod{x_j}$, so every solution x is congruent modulo n to exactly one of the d particular solutions $x_1, \ldots, x_{(d-1)}$. $\square$

Exercises for §2.1.

EXERCISE 2.1.1. Determine whether 3 and 99 are congruent modulo 7 or not.

EXERCISE 2.1.2. Show that if x is an odd integer, then $x^2 \equiv 1 \pmod 8$.

EXERCISE 2.1.3. Show that if $a, b \in \mathbb{Z}$ and $m, n \in \mathbb{N}$ are such that $n \mid m$ and $a \equiv b \pmod m$, then $a \equiv b \pmod n$.

EXERCISE 2.1.4. Show that if $n, k \in \mathbb{N}$ and $\{a_1, \ldots, a_k, b_1, \ldots, b_k\} \subset \mathbb{Z}$ satisfy $a_i \equiv b_i \pmod n$ for $i = 1, 2, ..., k$, then $\sum_{i=1}^{k} a_i \equiv \sum_{i=1}^{k} b_i \pmod n$.

EXERCISE 2.1.5. For which $n \in \mathbb{N}$ is it true that $1 + 2 + ... + (n-1) \equiv 0 \pmod n$?

2.2. Linear Congruences

Because congruence is analogous to equality, it is natural to ask about the analogues of linear equations, the simplest equations one can solve in algebra, but using congruence rather than equality. In this section, we discuss linear congruences of one variable and their solutions.

We start with a definition:

DEFINITION 2.2.1. Given constants $a, b \in \mathbb{Z}$ and $n \in \mathbb{Z}$, a congruence of the form $ax \equiv b \pmod{n}$ where $x \in \mathbb{Z}$ is unknown is called a **linear congruence in one variable**.

If a linear congruence has one solution, then it has infinitely many:

THEOREM 2.2.2. *Given constants $a, b \in \mathbb{Z}$, $n \in \mathbb{Z}$, and a solution $x \in \mathbb{Z}$ to the linear congruence $ax \equiv b \pmod{n}$, any other $x' \in \mathbb{Z}$ satisfying $x' \equiv x \pmod{n}$ is also a solution of the same congruence.*

[Note: in the early history of number theory, before Gauss, one talked about

DEFINITION 2.2.3. An algebraic equation whose constants and variables are all integers is called a **Diophantine equation**.

Then the modern linear congruence $ax \equiv b \pmod{n}$, for $a, b \in \mathbb{Z}$ and $n \in \mathbb{N}$ is equivalent to the linear Diophantine equation $ax - ny = b$ in the two unknowns x and y.]

The following gives a fairly complete characterization of solutions of linear congruences:

THEOREM 2.2.4. *Let $a, b \in \mathbb{Z}$ and $n \in \mathbb{N}$ and consider the linear congruence*

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

Setting $d = \gcd(a, n)$, we have

(1) *If $d \nmid b$, then the congruence has no solutions.*
(2) *If $d \mid b$, then the congruence has exactly d solutions which are distinct modulo n.*

PROOF. For Part 1, we prove its contrapositive. So assume the congruence has solutions, meaning $\exists x, k \in \mathbb{Z}$ such that $ax - b = kn$, or $ax - kn = b$. But since $d = \gcd(a, n)$ is a common divisor of a and n, it divides the linear combination $ax - kn = b$. Hence $d \mid b$.

Now for Part 2, assume $d \mid b$, so $\exists k \in \mathbb{Z}$ such that $kd = b$. But from Theorem 1.5.8 we know $\exists p, q \in \mathbb{Z}$ such that $d = pa + qn$. This means that $kpa + kqn = kd = b$ or, rearranging, $a(kp) - b = (-kq)n$. Hence $n \mid a(kp) = b$, *i.e.*, $a(kp) \equiv b \pmod{n}$ and thus $x = kp$ is one solution to the linear congruence $ax \equiv b \pmod{n}$.

Finally, let us prove that there are the correct number of solutions, mod n, of the congruence equation. We have just seen that there is at least one $x \in \mathbb{Z}$ satisfying $ax \equiv b \pmod{n}$. Let $y \in \mathbb{Z}$ be any other solution. Then $ax \equiv b \equiv ay \pmod{n}$. By part (2) of

Theorem 2.1.7, $x \equiv y \pmod{n/d}$, or $\delta = y - x \equiv 0 \pmod{n/d}$. Now by Theorem 2.1.9, there are exactly d possibilities, modulo n, for this δ. Thus there are d solutions of $ax \equiv b \pmod{n}$ of the form $x + \delta$. □

REMARK 2.2.5. Notice that if $a \in \mathbb{Z}$ and $n \in \mathbb{N}$ are relatively prime, then $\forall b \in \mathbb{Z}$ there is a unique solution modulo n to the equation $ax \equiv b \pmod{n}$.

EXAMPLE 2.2.6. Let us find all the solutions of the congruence $3x \equiv 12 \pmod{6}$. Notice that $\gcd(3, 6) = 3$ and $3 \mid 12$. Thus there are three incongruent solutions modulo 6. Using the Euclidean Algorithm to find the solution of the equation $3x - 6y = 12$ we get a solution $x_0 = 6$. Thus the three incongruent (modulo 6) solutions are given by $x_1 = 6 \pmod{6}$, $x_1 = 6 + 2 = 2 \pmod{6}$ and $x_2 = 6 + 4 = 4 \pmod{6}$.

As we mentioned in Remark 2.2.5, the congruence $ax \equiv b \pmod{n}$ for $a, b \in \mathbb{Z}$ and $n \in \mathbb{N}$ has a unique (modulo n) solution if $\gcd(a, n) = 1$. This will allow us to talk about *modular inverses*.

DEFINITION 2.2.7. Given $a \in \mathbb{Z}$ and $n \in \mathbb{N}$, a solution to the congruence $ax \equiv 1 \pmod{n}$ for $(a, n) = 1$ is called the **inverse of a modulo n**. We denote such an inverse by a^{-1}, with the n to be understood from context.

Stating formally what was just recalled from Remark 2.2.5, we have

COROLLARY 2.2.8. *Given $a \in \mathbb{Z}$ and $n \in \mathbb{N}$ which are relatively prime, the modular inverse a^{-1} exists and is unique modulo n.*

EXAMPLE 2.2.9. The modular inverse $7^{-1}$of 7 modulo 48 is 7. Notice that a solution of $7x \equiv 1 \pmod{48}$ is $x \equiv 7 \pmod{48}$.

Exercises for §2.2.

EXERCISE 2.2.1. Find all solutions of $3x \equiv 6 \pmod{9}$.

EXERCISE 2.2.2. Find all solutions of $3x \equiv 2 \pmod{7}$.

EXERCISE 2.2.3. Find inverses modulo 13 of 2 and of 11.

EXERCISE 2.2.4. Given $a \in \mathbb{Z}$ and $n \in \mathbb{N}$, show that if a^{-1} is the inverse of a modulo n and b^{-1} is the inverse of b modulo n, then $a^{-1}b^{-1}$ is the inverse of ab modulo n.

2.3. The Chinese Remainder Theorem

In this section, we discuss solutions of systems of congruences having different moduli. An example of this kind of systems is the following: find a number that leaves a remainder of 1 when divided by 2, a remainder of 2 when divided by three and a remainder of 3 when divided by 5. We shall see that there is a systematic way of solving this kind of system.

THEOREM 2.3.1. ***The Chinese Remainder Theorem:*** *Fix a $k \in \mathbb{N}$. Then given $b_1, \ldots, b_k \in \mathbb{Z}$ and $n_1, \ldots, n_k \in \mathbb{N}$, the system of congruences*

$$\begin{aligned} x &\equiv b_1 \pmod{n_1} \\ x &\equiv b_2 \pmod{n_2} \\ &\vdots \\ x &\equiv b_k \pmod{n_k} \end{aligned}$$

has a solution $x \in \mathbb{Z}$ if the $n_1, n_2, \ldots, n_k$ are pairwise relatively prime. The solution is unique modulo $N = n_1\, n_2 \ldots n_k$.

PROOF. For $j = 1, \ldots, k$, let $N_j = N/n_j$. Since the moduli n_j are pairwise relatively prime, $\gcd(N_j, n_j) = 1$ – after all, N_j is the product of all the moduli except n_j. Hence by Corollary 2.2.8, $\exists y_j = N_j^{-1}$ modulo n_j, satisfying $N_j y_j \equiv 1 \pmod{n_j}$. Consider now

$$x = \sum_{j=1}^{k} b_j N_j y_j$$

Since

$$N_j \equiv 0 \pmod{n_i} \ \forall i \neq j,$$

we see that

$$x \equiv b_j N_j y_j \equiv b_j \pmod{n_j}.$$

Hence x is a solution to the system of congruences.

We have to show now that any two solutions are congruent modulo N. Suppose that x and y are both solutions of the system of congruences. Then $x \equiv b_j \equiv y \pmod{n_j}$, or $n_j \mid x - y$, for all $1 \leq j \leq k$. But then, since the moduli are pairwise relatively prime, using Theorem 2.1.4 k times (formally, this needs to be done by induction!), we can conclude that $N = n_1 \ldots n_k \mid x - y$ or $x \equiv y \pmod{N}$. □

EXAMPLE 2.3.2. Solve the system

$$\begin{aligned} x &\equiv 1 \pmod{2} \\ x &\equiv 2 \pmod{3} \\ x &\equiv 3 \pmod{5}. \end{aligned}$$

We have $N = 2 \cdot 3 \cdot 5 = 30$. Also

$$N_1 = 30/2 = 15,\ N_2 = 30/3 = 10,\ \text{and } N_3 = 30/5 = 6.$$

So we have to solve now $15y_1 \equiv 1 \pmod 2$ – a solution is $y_1 \equiv 1 \pmod 2$. In the same way, we find that $y_2 \equiv 1 \pmod 3$ and $y_3 \equiv 1 \pmod 5$. Therefore

$$x = 1 \cdot 15 \cdot 1 + 2 \cdot 10 \cdot 1 + 3 \cdot 6 \cdot 1 = 53 \equiv 23 \pmod{30}.$$

Exercises for §2.3.

EXERCISE 2.3.1. Find an integer that leaves a remainder of 2 when divided by either 3 or 5, but that is divisible by 4.

EXERCISE 2.3.2. Find all integers that leave a remainder of 4 when divided by 11 and leaves a remainder of 3 when divided by 17.

EXERCISE 2.3.3. Find all integers that leave a remainder of 1 when divided by 2, a remainder of 2 when divided by 3 and a remainder of 3 when divided by 5.

EXERCISE 2.3.4. A band of 17 pirates steal some gold bars. When they try to divide the spoils equally, 3 bars are left over – so a fight breaks out, killing one. This immediately brings calm as they see if the gold can now be evenly shared. Unfortunately, there are now 10 bars left out, so they fight again. After the inevitable (single) further death, a perfect division is now possible. What is the minimum number of gold bars the pirates could have started with? *[This is apparently an ancient Chinese problem.]*

2.4. Another Way to Work with Congruences: Equivalence Classes

In this section, we shall consider another way to work with congruences, based upon the following:

DEFINITION 2.4.1. Let S be a set and $\cong$ a *relation* defined on S. (That is, $\forall x, y \in S$, the statement "$x \cong y$" may be true or false.) If $\cong$ satisfies the following three properties, it is called an **equivalence relation**:

- *[Reflexivity]* $\forall x \in S$, $x \cong x$.
- *[Symmetry]* $\forall x, y \in S$, $x \cong y \Leftrightarrow y \cong x$.
- *[Transitivity]* $\forall x, y, z \in S$, $x \cong y$ and $y \cong z \Rightarrow x \cong z$.

If $\cong$ is an equivalence relation on the set S and $x \in S$, then the set $[x] = \{y \in S \mid y \cong x\} \subseteq S$ is called the **equivalence class of** x. We write $S/\cong$ for the set of equivalence classes in S. And if $\mathcal{C} \in S$, then any $r \in S$ such that $\mathcal{C} = [r]$ is called a **representative of the equivalence class** $\mathcal{C}$.

THEOREM 2.4.2. *Let S be a set and $\cong$ an equivalence relation defined on S. Then*

(1) $\forall x \in S,\ x \in [x]$.

(2) $\forall x, y \in S$, *either* $[x] = [y]$ *or* $[x] \cap [y] = \emptyset$, *but not both.*

PROOF. (1): This is just the reflexivity of $\cong$.

(2): Suppose $z \in [x] \cap [y]$. This means $z \cong x$ and $z \cong y$, so by symmetry $y \cong z$ and by transitivity, $x \cong y$.

Now if $a \in [x]$ and $b \in [y]$, then $a \cong x$ and $b \cong y$. By transitivity, $a \cong x \cong y \cong b$. Thus $a \in [y]$ and, by symmetry, $b \cong x$ so $b \in [x]$.

Therefore $[x] \subseteq [y]$ and $[y] \subseteq [x]$, and thus $[x] = [y]$.

Since the above construction only relied on the existence of some element $z \in [x] \cap [y]$, we see that the only way we could fail to have $[x] = [y]$ is if $[x] \cap [y] = \emptyset$. □

EXAMPLE 2.4.3. On the set $S = \{(n, m) \mid n, m \in \mathbb{Z}, m \neq 0\}$ we can define the relation $(a, b) \cong (c, d)$ if $ad = cb$. Then $S/\cong$ is nothing other than the rational numbers, $\mathbb{Q}$!

Now let us specialize the concept of equivalence class to the case of congruences:

PROPOSITION 2.4.4. *Given $n \in \mathbb{N}$, the relation on $\mathbb{Z}$ defined by*

$$a \cong_n b \Leftrightarrow a \equiv b \pmod{n}$$

(which we shall write $a \cong b$ if the n is clear from context) is an equivalence relation.

DEFINITION 2.4.5. Given $n \in \mathbb{N}$ and $a \in \mathbb{Z}$, the equivalence class of a under the above equivalence relation is called the **congruence class of** a **mod** n and is written $[a]_n$ (or, by abuse of notation, merely $[a]$ if the n is understood from the context). The set of

equivalence classes $\mathbb{Z}/\cong_n$ is called the **integers mod** n and is written $\mathbb{Z}/n\mathbb{Z}$ (or, by some authors, $\mathbb{Z}/n$ or $\mathbb{Z}_n$).

THEOREM 2.4.6. *Given $n \in \mathbb{N}$, $\mathbb{Z}/n\mathbb{Z}$ has n elements, with representatives $0, \ldots, n-1$ of these distinct equivalence classes. That is,*

$$\mathbb{Z}/n\mathbb{Z} = \{[0]_n, \ldots, [n-1]_n\}.$$

PROOF. Given $n \in \mathbb{N}$ and $a \in \mathbb{Z}$, the Division Algorithm tells us there is a unique pair $q, r \in \mathbb{Z}$ such that $a = qn + r$ and $0 \leq r < n$. Note that $a \equiv r \pmod{n}$ or, equivalently, $a \in [r]$. Thus every $a \in \mathbb{Z}$ is an element of a unique equivalence class $[r]$ for $r \in \{0, \ldots, n-1\}$. Since each such $r \in [r]$, uniquely!, the equivalence classes $[0], \ldots, [n-1]$ are all distinct. □

The remarkable thing about these $\mathbb{Z}/n\mathbb{Z}$ is that we can do much of the usual integer arithmetic in them; in fact, sometimes we can do a bit more than the usual.

DEFINITION 2.4.7. Given $n \in \mathbb{N}$ and $\mathcal{C}, \mathcal{D} \in \mathbb{Z}/n\mathbb{Z}$, define $\mathcal{C} + \mathcal{D} = [a + b]$ and $\mathcal{C} \cdot \mathcal{D} = [a \cdot m]$ where a and b are any representatives of the congruence classes $\mathcal{C}$ and $\mathcal{D}$, respectively.

THEOREM 2.4.8. *The operations $+$ and $\cdot$ on $\mathbb{Z}/n\mathbb{Z}$ are well-defined. That is, they are independent of the choices of representatives of the congruence classes.*

PROOF. Say $n \in \mathbb{N}$ and $\mathcal{C}, \mathcal{D} \in \mathbb{Z}/n\mathbb{Z}$. Let $a, p, b, q \in \mathbb{Z}$ be such that $\mathcal{C} = [a] = [p]$ and $\mathcal{D} = [b] = [q]$. Then $p \equiv a \pmod{n}$ and $q \equiv b \pmod{n}$. But then Theorem 2.1.3 tells us that $a + b \equiv p + q \pmod{n}$ and $a \cdot b \equiv p \cdot q \pmod{n}$, so $[a + b] = [p + q]$ and $[a \cdot b] = [p \cdot q]$. This means that $\mathcal{C} + \mathcal{D}$ can be defined either as $[a + b]$ or as $[p + q]$ and it will be the same thing, as asserted, and likewise for $\mathcal{C} \cdot \mathcal{D}$. □

These new operations are quite nice:

THEOREM 2.4.9. *Given $n \in \mathbb{N}$, the addition and multiplication on $\mathbb{Z}/n\mathbb{Z}$*

(1) *are commutative and associative;*
(2) *multiplication distributes over addition;*
(3) *both operations have an identity – $[0]$ for addition and $[1]$ for multiplication;*
(4) *every element of $\mathbb{Z}/n\mathbb{Z}$ has an additive inverse – the inverse of $[a]$ is $[-a]$ (or $[n - a]$, another name for the same thing); and*
(5) *an element $[a] \in \mathbb{Z}/n\mathbb{Z}$ has a multiplicative inverse $[a]^{-1}$ if and only if $\gcd(a, n) = 1$; this inverse is unique when it exists.*

PROOF. Left to the reader. Note the last point is basically Corollary 2.2.8 restated in the language of congruence classes. □

We can also restate many of our other, earlier results beside just Corollary 2.2.8 in terms of congruence classes. Most of these shall be left to the reader, but here is one example:

THEOREM 2.4.10. *Given $a, b \in \mathbb{Z}$ and $n \in \mathbb{N}$, let $d = \gcd(a, n)$. Consider the equation*

$$[a] \cdot x = [b]$$

where $x \in \mathbb{Z}/n\mathbb{Z}$. Then

(1) *If $d \nmid b$, there are no solutions.*
(2) *If $d \mid b$, this equation has exactly d solutions in $\mathbb{Z}/n\mathbb{Z}$.*

PROOF. Left to the reader; this is really just Theorem 2.2.4 stated differently. □

Exercises for §2.4.

EXERCISE 2.4.1. When the rational numbers $\mathbb{Q}$ are described as in Example 2.4.3, we define addition by $[(n,m)] + [(p,q)] = [(nq+mp, mq)]$ and multiplication by $[(n,m)] \cdot [(p,q)] = [(np, mq)]$, where $n, m, p, q \in \mathbb{Z}$ and neither m nor q is 0. Prove a version of Theorems 2.4.8 for this version of $\mathbb{Q}$.

What are the additive and multiplicative identities in this $\mathbb{Q}$? Does every element (or nearly every element) of $\mathbb{Q}$ have an additive and multiplicative inverse – if so, give a formula for those inverses; if not, why not?

EXERCISE 2.4.2. Restate the Chinese Remainder Theorem in terms of congruence classes and *equations* in various $\mathbb{Z}/n\mathbb{Z}$'s rather than congruences.

EXERCISE 2.4.3. Prove the statements in this section which have proofs "left to the reader."

EXERCISE 2.4.4. State and prove congruence class versions of any results about congruences from sections 2.1 and 2.2 which do not have versions in this section.

2.5. Euler's ϕ Function

Euler made the following definition, and it was good.

DEFINITION 2.5.1. Given $n \in \mathbb{N}$,

$$\phi(n) = \#\left(\{m \in \mathbb{Z} \mid 0 \leq m < n \text{ and } \gcd(m,n) = 1\}\right) .$$

In other words, $\phi(n)$ counts the number of non-negative integers less than n which are relatively prime to n.

This is called **Euler's ϕ function**, or **Euler's totient function** ("totient" rhymes with "quotient"; this name was give to it by the English mathematician Sylvester).

Here's one thing it is good for:

THEOREM 2.5.2. *Given $n \in \mathbb{N}$, $\phi(n)$ is the number of elements of $\mathbb{Z}/n\mathbb{Z}$ which are (multiplicatively) invertible.*

PROOF. This is just Theorem 2.4.9 part (5) □

One quite surprising fact about Euler's totient function is that it is multiplicative, at least for relatively prime numbers:

THEOREM 2.5.3. *Given $n, m \in \mathbb{Z}$, if $\gcd(n,m) = 1$ then $\phi(nm) = \phi(n)\phi(m)$.*

PROOF. This is actually a nice application of the Chinese Remainder Theorem, as we shall see.

Fix $n, m \in \mathbb{N}$ which are relatively prime. For $k \in \mathbb{N}$, let

$$(\mathbb{Z}/k\mathbb{Z})^* = \{x \in \mathbb{Z}/k\mathbb{Z} \mid x \text{ is invertible}\}$$

so $\phi(k) = \#((\mathbb{Z}/k\mathbb{Z})^*)$. (This notation means *the number of things in the set* $(\mathbb{Z}/k\mathbb{Z})^*$.)

Now define the set of pairs

$$(\mathbb{Z}/n\mathbb{Z})^* \times (\mathbb{Z}/m\mathbb{Z})^* = \{(a,b) \mid a \in (\mathbb{Z}/n\mathbb{Z})^* \text{ and } b \in (\mathbb{Z}/n\mathbb{Z})^*\} .$$

Notice that $\#((\mathbb{Z}/n\mathbb{Z})^* \times (\mathbb{Z}/m\mathbb{Z})^*) = \#((\mathbb{Z}/n\mathbb{Z})^*) \cdot \#((\mathbb{Z}/m\mathbb{Z})^*) = \phi(n)\phi(m)$, since each part of the pair is free to be whichever element it wants of the respective set, so the total number of pairs is the size of the first set times the size of the second. Therefore, if we can prove that $(\mathbb{Z}/n\mathbb{Z})^* \times (\mathbb{Z}/m\mathbb{Z})^*$ can be put into bijective correspondence with $(\mathbb{Z}/(nm)\mathbb{Z})^*$, then we will have

$$\phi(n)\phi(m) = \#\left((\mathbb{Z}/n\mathbb{Z})^* \times (\mathbb{Z}/m\mathbb{Z})^*\right) = \#((\mathbb{Z}/(nm)\mathbb{Z})^*) = \phi(nm)$$

as desired, since bijective sets have the same number of elements.

The correspondence is given by the function

$$\mathcal{F} : (\mathbb{Z}/(nm)\mathbb{Z})^* \to (\mathbb{Z}/n\mathbb{Z})^* \times (\mathbb{Z}/m\mathbb{Z})^* : [x]_{nm} \mapsto ([x]_n, [x]_m) .$$

We must show $\mathcal{F}$ is well-defined, 1-1, and onto. Well-defined means that for some $[x]_{nm}$, if $y \in \mathbb{Z}$ is another representative of the congruence class $[x]_{nm}$, then $([x]_n, [x]_m) = ([y]_n, [y]_m)$ so that $\mathcal{F}$ is defined only by the class $[x]_{nm}$, not by its choice of representative x. But this is easy, since $y \in [x]_{nm}$ means $y \cong_{nm} x$ so $\exists k \in \mathbb{Z}$ such that $y - x = k(nm) = (km)n = (kn)m$. These last two versions mean that $y \cong_n x$ and $y \cong_m x$, so $[x]_n = [y]_n$ and $[x]_m = [y]_m$, which means $\mathcal{F}$ is well-defined.

Now let us assume that $x, y \in \mathbb{Z}$ have $\mathcal{F}([x]_{nm}) = \mathcal{F}([y]_{nm})$, *i.e.,* $[x]_n = [y]_n$ and $[x]_m = [y]_m$. That is, $z = x - y \in \mathbb{Z}$ solves the system

$$\begin{aligned} z &\equiv 0 \pmod{n} \\ z &\equiv 0 \pmod{m}\,. \end{aligned}$$

But $z = 0$ also solves this system, and the Chinese Remainder Theorem tells us that solutions of this system (which is an appropriate system for the CRT since $\gcd(n, m) = 1$) are unique modulo nm. Therefore, $x - y \equiv 0 \pmod{nm}$, so $x \equiv y \pmod{nm}$. Hence $[x]_{nm} = [y]_{nm}$, and thus $\mathcal{F}$ is 1-1.

Now, given any pair $([x]_n, [y]_m) \in (\mathbb{Z}/n\mathbb{Z})^* \times (\mathbb{Z}/m\mathbb{Z})^*$, consider the system of congruences

$$\begin{aligned} z &\equiv x \pmod{n} \\ z &\equiv y \pmod{m}\,, \end{aligned}$$

which system is again CRT-ready since $\gcd(n, m) = 1$. Therefore there exists a solution to this system, call it $z \in \mathbb{Z}$, and $\mathcal{F}([z]_{nm}) = ([x]_n, [y]_m)$. Thus $\mathcal{F}$ is onto as well. $\square$

Exercises for §2.5.

EXERCISE 2.5.1. Compute $\phi(n)$ for $n = 2, 3, 5, 7, 11, 13, 17$. Make a general conjecture. Can you prove it?

EXERCISE 2.5.2. Compute $\phi(n)$ for $n = 2, 4, 8, 16, 32, 64$. Make a conjecture about $\phi(2^k)$ for $k \in \mathbb{N}$. Prove it!

CHAPTER 3

Prime Numbers

Primes are the atoms out of which the more complicated, composite integers (the molecules, in this metaphor) are built. In this chapter we study some of their basic properties, prove the aptly named Fundamental Theorem of Arithmetic, and go on to Wilson's Theorem.

3.1. Basics and the FTA

First of all, we make the

DEFINITION 3.1.1. We say $p \in \mathbb{N}$ is **prime** if $p > 1$ and the only natural numbers which divide p are 1 and p.

EXAMPLE 3.1.2. Some primes are 2, 3, 5, 7, 11, 13, and 17. Notice 2 is the only even prime (clearly – any other would be a multiple of 2 and hence could not be prime), and has some unusual properties – the joke is that "2 is the oddest prime."

The largest prime known to humans at the time of this writing is

$$2^{57,885,161} - 1$$

which was proven to be prime in January of 2013 by a distributed computer program called **GIMPS** [the *Great Internet Mersenne Prime Search*] running on hundreds of machines across the Internet.

In contrast, we also use the following term

DEFINITION 3.1.3. A number $c \in \mathbb{N}$ which is greater than 1 and not prime is called **composite**.

How far does a naive, brute-force check have to go in order to see if a number is composite?

THEOREM 3.1.4. *If n is a composite, then it has a positive divisor d satisfying $d \leq \sqrt{n}$.*

PROOF. Suppose n is composite. Then it has some divisor $a \in \mathbb{N}$. Notice that $n = a \cdot \frac{n}{a}$, so $\frac{n}{a} \in \mathbb{N}$ is also a divisor. But a and $\frac{n}{a}$ cannot both be less than $\sqrt{n}$, because if they were we would have

$$n = a \cdot \frac{n}{a} < \sqrt{n} \cdot \sqrt{n} = n$$

which would be a contradiction. Hence either a or $\frac{n}{a}$ is the divisor d promised by the theorem statement. □

Euclid's Lemma (Lemma 2.1.6) takes a particularly nice form if the divisor involved is prime:

PROPOSITION 3.1.5. *Suppose p is a prime and $a, b \in \mathbb{Z}$. If $p \mid ab$ then $p \mid$ or $p \mid b$.*

PROOF. Notice that $\gcd(p, a) \mid p$, therefore $\gcd(p, a)$ is either 1 or p since p is prime. But also $\gcd(a, p) \mid a$, so either $p \mid a$ or $\gcd(a, p) = 1$. If $p \mid a$, we are done. If not, since therefore $\gcd(a, p) = 1$, Euclid's Lemma 2.1.6 tells us that $p \mid b$. □

A more general form of this is

COROLLARY 3.1.6. *Suppose p is a prime, $k \in \mathbb{N}$, and $a_1, \ldots, a_k \in \mathbb{Z}$. Then if $p \mid a_1 \ldots a_k$, it follows that p divides at least one of the a_j.*

PROOF. Left to the reader (use induction on k). □

This leads to the aptly named

THEOREM 3.1.7. ***The Fundamental Theorem of Arithmetic:*** *Let $n \in \mathbb{N}$, $n \geq 2$. Then $\exists k \in \mathbb{N}$ and primes $p_1, \ldots, p_k$ such that $n = p_1 \ldots p_k$. Furthermore, if $l \in \mathbb{N}$ and $q_1, \ldots, q_l$ are also primes such that $n = q_1 \ldots q_l$, then $l = k$ and the factorization in terms of the q's is merely a reordering of that in term s of the p's.*

PROOF. We use the Second Principle of Mathematical induction for the existence part. The general statement we are proving is $\forall n \in \mathbb{Z},\ n > 1 \Rightarrow S(n)$, where $S(n)$ is the statement "$\exists k \in \mathbb{N}$ and primes $p_1, \ldots, p_k$ such that $n = p_1 \ldots p_k$."

As the base case, say $n = 2$. Then $k = 1$ and $p_1 = 2$ works.

Now assume $S(k)$ is true for all $k < n$. If n is prime, then $k = 1$ and $p_1 = n$ works. So suppose n is instead composite, with divisor $d \neq 1, n$. Then both $d, \frac{n}{d} < n$, so by the inductive hypothesis $\exists k, k' \in \mathbb{N}$ and primes $p_1, \ldots, p_k, p'_1, \ldots, p'_{k'}$ such that $d = p_1 \ldots p_k$ and $\frac{n}{d} = p'_1 \ldots p'_{k'}$. So then n is the product

$$n = p_1 \ldots p_k \cdot p'_1 \ldots p'_{k'}$$

of the $k + k'$ primes. Hence $S(n)$ is true as well, and so prime factorizations always exist.

Now suppose $n \in \mathbb{Z}$ satisfies $n > 1$ and $\exists k, l \in \mathbb{N}$ and both primes $p_1, \ldots p_k$ and $q_1, \ldots, q_l$ such that

$$p_1 \ldots p_k = n = q_1 \ldots q_l \ .$$

Certainly p_1 divides the left hand side of these dual expressions for n. Then by Corollary 3.1.6, p_1 divides one of the q_j, which means it must be that $p_1 = q_j$ since they are prime. Removing the p_1 from the left and the q_j from the right, we get

$$p_2 \ldots p_k = n = q_1 \ldots q_{j-1} \cdot q_{j+1} \ldots q_l \ .$$

Continuing in this way, either we get the uniqueness statement in the theorem, or we run out of p's or q's. However, we cannot run of of primes on one side before the other, because that would make a product of primes on one side equal to 1, which is impossible. $\square$

Exercises for §3.1.

EXERCISE 3.1.1. Provide all the details of the proof of Corollary 3.1.6.

EXERCISE 3.1.2. State and prove a theorem about the prime factorizations of numbers $a, b \in \mathbb{N}$ and of their gcd.

EXERCISE 3.1.3. A number $n \in \mathbb{Z},\ n > 1$ is called *square-free* if it is not divisible by the square of any natural number other than 1. Prove that an $n \in \mathbb{Z},\ n \geq 1$ is square-free if and only if it is the product of distinct primes.

3.2. Wilson's Theorem

In this section, we prove a nice theorem usually named after an 18^{th} century English mathematician ... although it was actually first stated by Ibn al-Haytham nearly 800 years earlier.

First, we need a

LEMMA 3.2.1. *Let p be a prime. Then $n \in \mathbb{N}$ equals its own inverse mod p if and only if $p \mid n+1$ or $p \mid n-1$, i.e., iff $n \equiv \pm 1 \pmod{p}$.*

PROOF. Say p is a prime and $n \in \mathbb{N}$ equals its own inverse mod p. That means that $n^2 = n \cdot n \equiv 1 \pmod{p}$. By definition, $p \mid n^2 - 1 = (n+1)(n-1)$. By Proposition 3.1.5, this means that either $p \mid n+1$ or $p \mid n-1$.

For the converse, suppose $n \equiv 1 \pmod{p}$ or $n \equiv -1 \pmod{p}$. Then, by Theorem 2.1.3, $n^2 \equiv (\pm 1)^2 = 1 \pmod{p}$, so n equals its own inverse mod p. □

This Lemma is a key step in

THEOREM 3.2.2. ***Wilson's Theorem*** *Given $p \in \mathbb{N}$ such that $p \geq 2$, p is prime iff $(p-1)! \equiv -1 \pmod{p}$.*

PROOF. Suppose p is prime. Consider the terms of $(p-2)!$: every one has an inverse mod p by Corollary 2.2.8, and only 1 equals its own inverse mod p (so would $p-1$, but it is not in the product $(p-2)!$ by Lemma 3.2.1. Hence in $(p-1)!$ we can group the terms into pairs ($(p-3)/2$ of these pairs) which are inverses mod p, leaving out only 1 and $(p-1)$. That is

$$(p-1)! \equiv 1^{(p-3)/2}(p-1) \equiv p-1 \equiv -1 \pmod{p}$$

Conversely, suppose $p \in \mathbb{N}$ satisfies $p \geq 2$ and $(p-1)! \equiv -1 \pmod{p}$. We can rewrite that congruence as $(p-1)! + 1 \equiv 0 \pmod{p}$, or $p \mid ((p-1)! + 1)$.

Now let $d \in \mathbb{N}$ be a divisor of p such that $d \neq p$. It follows that d is one of the numbers in the product $(p-1)!$, so $d \mid (p-1)!$; also, since $d \mid p$ and $p \mid ((p-1)! + 1)$, it follows that $d \mid ((p-1)! + 1)$. Therefore, $d \mid ((p-1)! + 1) - (p-1)! = 1$ by Theorem 1.3.9, which means $d = 1$.

In other words, if $d \in \mathbb{N}$ is a divisor of p, it must be either p or 1, so p is prime. □

EXAMPLE 3.2.3. Let's work through one direction of the proof for a simple case, say of $p = 7$. We start by finding all the inverses of the numbers $2, \ldots, 6$ (by brute force, in this small case): $2 \cdot 4 \equiv 1 \pmod{7}$ and $3 \cdot 5 \equiv 1 \pmod{7}$, meaning that $2^{-1} \equiv 4$ (or$4^{-1} \equiv 2$) and $3^{-1} \equiv 5$ (or $5^{-1} \equiv 3$).

Then

$$(p-2)! = 5! = 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = (5 \cdot 3) \cdot (4 \cdot 2) \equiv 1 \cdot 1 \equiv 1 \pmod{7}$$

which in turn means that

$$(p-1)! = 6! = 6 \cdot 5! \equiv 6 \cdot 1 \equiv 6 \equiv 7 - 1 \equiv -1 \pmod{7}.$$

Notice that Wilson's Theorem can be used to build a test for primality: see if a number n satisfies $(n-1)! \equiv -1 \pmod{n}$ and, if so, n is prime. This is an entirely impractical test, but it is our first example of a simple computational congruence involving an integer which can tell us that that integer is prime.

3.3. Multiplicative Order and Applications

In this section we prove two very useful results called Euler's Theorem and Fermat's Little Theorem (a special case of Euler's). We do not follow the proof strategy of Euler and Fermat, however, instead using an approach inspired by abstract algebra and Lagrange's Theorem in that subject.

First we need the

DEFINITION 3.3.1. Suppose $n \in \mathbb{N}$ and $a \in \mathbb{Z}$ satisfy $n \geq 2$ and $\gcd(a, n) = 1$. Then we define the **multiplicative order of a in mod n** (called just the *order* when the *multiplicative* and n can be understood from context) to be the smallest $k \in \mathbb{N}$ such that $a^k \equiv 1 \pmod{n}$. The order of a in mod n is written $\mathrm{ord}_n(a)$.

Let us verify something which should probably always be checked for a new definition:

PROPOSITION 3.3.2. *Given relatively prime numbers $n \in \mathbb{N}$ and $a \in \mathbb{Z}$ with $n \geq 2$, $\mathrm{ord}_n(a)$ is well-defined.*

PROOF. The problem in the definition of $\mathrm{ord}_n(a)$ might be that there might not be any value of $k \in \mathbb{N}$ at all for which $a^k \equiv 1 \pmod{n}$.

But notice that this is a congruence, so we are really only concerned with the elements a^k up to their congruence class.

So look at $\{[a^p]_n \mid p \in \mathbb{N}\}$ and imagine putting the natural numbers $p \in \mathbb{N}$ into different boxes based on what is the corresponding congruence class $[a^p] \in \mathbb{Z}/n\mathbb{Z}$. Since there are infinitely many elements in $\mathbb{N}$ and only n elements in $\mathbb{Z}/n\mathbb{Z}$ – n *boxes* – by the Pigeonhole Principle (Theorem 1.1.2) there must be two (actually, infinitely many pairs of) distinct values $p, q \in \mathbb{N}$ such that p and q end up in the same box, meaning $[a^p] = [a^q]$. Assume without loss of generality that $p > q$, so $p - q \in \mathbb{N}$.

We are given that $\gcd(a, n) = 1$, so by Corollary 2.2.8, a^{-1} exists in mod n. Thus

$$a^{p-q} \equiv a^p \, (a^{-1})^q \equiv a^q \, (a^{-1})^q \equiv a^0 \equiv 1 \pmod{n}$$

(replacing a^p with a^q in the middle of this congruence since we know that $[a^p] = [a^q]$, which means $a^p \equiv a^q \pmod{n}$) and therefore the set of $k \in \mathbb{N}$ for which $a^k \equiv 1 \pmod{n}$ is non-empty. The order is the smallest such value, which exists by the Well-Ordering Principle. □

Here now is a theorem from abstract algebra (Lagrange's Theorem) translated into the current context:

THEOREM 3.3.3. *Given relatively prime numbers $n \in \mathbb{N}$ and $a \in \mathbb{Z}$, $\mathrm{ord}_n(a) \mid \phi(n)$.*

PROOF. Start by looking at the congruence classes $[a], [a^2], [a^3] \ldots$. They go up to $[a^{\mathrm{ord}_n(a)}] = [1]$ and then start to repeat, so the set

$$\langle a \rangle = \{[a^j] \mid j \in \mathbb{N}, j \leq \mathrm{ord}_n(a)\}$$

is a set of $\mathrm{ord}_n(a)$ elements of $\mathbb{Z}/n\mathbb{Z}$ (in group theory, this set $\langle a\rangle$ is called the *cyclic subgroup of* $(\mathbb{Z}/n\mathbb{Z})^*$ *generated by* a). Notice that because $a^{\mathrm{ord}_n(a)} \equiv 1 \pmod n$, $[1] \in \langle a\rangle$. Also, since if a^{-1} is the inverse of a mod n, $(a^{-1})^k$ is the inverse of a^k for $k \in \mathbb{N}$, we see that $\langle a\rangle \subseteq (\mathbb{Z}/n\mathbb{Z})^*$.

To finish, we shall show that $(\mathbb{Z}/n\mathbb{Z})^*$ is made up of some number, say m, of pieces, which we shall call *cosets*, each of which is bijective with $\langle a\rangle$. These pieces will thus all have $\mathrm{ord}_n(a)$ elements, so $m \cdot \mathrm{ord}_n(a) = \#\left((\mathbb{Z}/n\mathbb{Z})^*\right) = \phi(n)$, from which our desired result follows.

A *coset of* $\langle a\rangle$ is a set of the form

$$x\langle a\rangle = \{[x]\cdot[a^j] = [x\,a^j] \mid j \in \mathbb{N}, j \le \mathrm{ord}_n(a)\}$$

where $[x] \in (\mathbb{Z}/n\mathbb{Z})^*$. We have observed that $[1] \in \langle a\rangle$, so $\forall [x] \in (\mathbb{Z}/n\mathbb{Z})^*$, $[x] \in x\langle a\rangle$, meaning that every congruence class $[x] \in (\mathbb{Z}/n\mathbb{Z})^*$ is in some coset, ***i.e.,***

$$\mathbb{Z}/n\mathbb{Z} \subseteq \bigcup_{[x]\in(\mathbb{Z}/n\mathbb{Z})^*} x\langle a\rangle \ .$$

In fact, each coset $x\langle a\rangle$ is bijective with $\langle a\rangle$. The bijection is the map $f_x : \langle a\rangle \to x\langle a\rangle$ defined by $f_x([a^j]) = [xa^j]$ for $j \in \mathbb{N}$ with $j \le \mathrm{ord}_n(a)$. This map is a bijection because it has an inverse $f_{x^{-1}}$, coming from the inverse mod n of x.

Now suppose $x\langle a\rangle$ and $y\langle a\rangle$ are two cosets. I claim that either $x\langle a\rangle = y\langle a\rangle$ or $x\langle a\rangle \bigcap y\langle a\rangle = \emptyset$. For this, suppose $x\langle a\rangle \bigcap y\langle a\rangle \neq \emptyset$, so $\exists [z] \in x\langle a\rangle \bigcap y\langle a\rangle$. That means that $\exists j,k \in \mathbb{N}$ such that $j \le \mathrm{ord}_n(a)$, $k \le \mathrm{ord}_n(a)$, and $[xa^j] = [z] = [ya^k]$. In other words,

$$xa^j \equiv ya^k \pmod n .$$

Without loss of generality, assume $j \le k$. Then if we multiply both sides of the above congruence by $(a^{-1})^j$, we have $x \equiv ya^{k-j} \pmod n$. But this means that every element $[xa^p] \in x\langle a\rangle$ is can be expressed as $[ya^{p+k-j}] \in y\langle a\rangle$, and every element $[ya^q] \in y\langle a\rangle$ is can be expressed as $[xa^{q+j-k}] \in x\langle a\rangle$. Thus $x\langle a\rangle = y\langle a\rangle$. □

That was a lot of work, but now we get the famous theorems of Fermat's Little and of Euler very easily.

THEOREM 3.3.4. ***Euler's Theorem*** *Say* $a \in \mathbb{Z}$ *and* $n \in \mathbb{N}$ *satisfy* $\gcd(a,n) = 1$. *Then* $a^{\phi(n)} \equiv 1 \pmod n$.

PROOF. The previous theorem, 3.3.3, told us that $\mathrm{ord}_n(a) \mid \phi(n)$, so $\exists m \in \mathbb{Z}$ such that $m \cdot \mathrm{ord}_n(a) = \phi(n)$. By the definition of order, this means

$$a^{\phi(n)} \equiv a^{m\,\mathrm{ord}_n(a)} \equiv \left(a^{\mathrm{ord}_n(a)}\right)^m \equiv 1^m \equiv 1 \pmod n.$$

□

COROLLARY 3.3.5. ***Fermat's Little Theorem*** *If p is a prime and $a \in \mathbb{Z}$ satisfies $\gcd(p, a) = 1$ then $a^{p-1} \equiv 1 \pmod{p}$.*

PROOF. We have seen that for primes p, $\phi(p) = p - 1$, so there is very little to do here. □

Sometimes one sees Fermat's Little Theorem in the following, different form:

THEOREM 3.3.6. *If p is a prime then $\forall a \in \mathbb{Z}$, $a^p \equiv a \pmod{p}$.*

PROOF. If $\gcd(a, p) = 1$, then by Fermat's Little, $a^{p-1} \equiv 1 \pmod{p}$. Multiplying both sides of this congruence by a yields the desired result.

If instead $\gcd(a, p) \neq 1$, it must be that $\gcd(a, p) = p$ since that gcd is a divisor of p, which is prime. The gcd is also a divisor of a, so in fact $p \mid a$. But then a and all its powers are congruent mod p to 0, so $a^p \equiv 0 \equiv a \pmod{p}$. □

Exercises for §3.3.

EXERCISE 3.3.1. What are the remainder when $15!$ is divided by 17 and the remainder when $2 \cdot (26!)$ is divided by 29?

EXERCISE 3.3.2. We know 17 is prime (it's just a wonderful number, isn't it?). But just to be sure, use Wilson's Theorem to prove it.

EXERCISE 3.3.3. Can we build a test for primality out of Fermat's Little Theorem? If so, would it be better (more efficient) than one based on Wilson's Theorem? How would it work? Write a clear, formal statement of your proposed primality test.

If you know a programming language, write some code to try out your primality test. If not (or, in any case, after the programming), do a little research to see if this question has already been investigated and, if so, what was the conclusion. Give either a formal statement of the test and formal result describing its efficacy or a counter-example to your test that you found in the literature.

EXERCISE 3.3.4. Suppose p is an odd prime. Prove that if the quadratic congruence $x^2 + 1 \equiv 0 \pmod{p}$ has a solution, then $p \equiv 1 \pmod{4}$. *[Hint: Apply Fermat's Little Theorem to a solution a of the congruence, then multiply and divide by 2 in the power.]*

3.4. Another Approach to Fermat's Little and Euler's Theorems

There is another way to think about these theorems which we shall explain here, because it usefully fills out our understanding of multiplication in $\mathbb{Z}/n\mathbb{Z}$.

In this case, we shall consider the theorems in the reverse order to what we used above – which is in fact more historically accurate.

THEOREM 3.4.1. ***Fermat's Little Theorem, Redux*** *If p is a prime and $a \in \mathbb{Z}$ satisfies $\gcd(p, a) = 1$ then $a^{p-1} \equiv 1 \pmod{p}$.*

PROOF. Let's start by proving that the set

$$M_a = \{[0]_p, [a]_p, [2a]_p, \ldots, [(p-1)a]_p\}$$

of multiples of $[a]_p$ in $\mathbb{Z}/p\mathbb{Z}$ has p elements.

There are only p congruence classes named between the $\{$ and $\}$, so the set cannot have more than p elements.

Now look at two elements in this set, call them $[ja]_p$ and $[ka]_p$ with $0 \leq j, k < p$, and suppose they are equal. That means that $ja \equiv ka \pmod{p}$, or $p \mid (j-k)a$. By Proposition 3.1.5, this means that either $p \mid (j-k)$ or $p \mid a$. Since $\gcd(p, a) = 1$, we cannot have $p \mid a$. Thus $p \mid (j-k)$.

But $0 \leq j, k < p$ tells us that $-p+1 < j-k < p-1$ and the only multiple of p in this range is 0. Therefore $j = k$, and this means that all of the elements in the above description of M_a are distinct, so indeed $\#(M_a) = p$. But we already know that $\mathbb{Z}/p\mathbb{Z}$ itself has p elements, which means M_a above is simply another way of describing $\mathbb{Z}/p\mathbb{Z}$.

For our next step, let's multiply together all the nonzero elements of M_a, or of $\mathbb{Z}/p\mathbb{Z}$ since we know it's the same thing:

$$a \cdot 2a \cdot \cdots \cdot (p-1)a \equiv 1 \cdot 2 \cdot \cdots \cdot (p-1) \pmod{p}.$$

Rearranging these terms, we see

$$a^{p-1}(p-1)! \equiv (p-1)! \pmod{p}$$

or $p \mid (a^{p-1} - 1)(p-1)!$. Since p being prime means $\gcd(p, (p-1)!) = 1$, it follows by Proposition 3.1.5 that $p \mid a^{p-1} - 1$. That is, $a^{p-1} \equiv 1 \pmod{p}$, as desired. □

The above is quite similar to the proof that Euler gave of Fermat's Little Theorem (actually, Fermat gave no proof at all – not unlike his famous Last "Theorem"). A very similar strategy can also prove his own theorem:

THEOREM 3.4.2. ***Euler's Theorem, Redux*** *Say $a \in \mathbb{Z}$ and $n \in \mathbb{N}$ satisfy $\gcd(a, n) = 1$. Then $a^{\phi(n)} \equiv 1 \pmod{n}$.*

PROOF. Recall that the set $(\mathbb{Z}/n\mathbb{Z})^*$ of (multiplicatively) invertible elements in $\mathbb{Z}/n\mathbb{Z}$ can be written as $\{[b_1]_n, \ldots, [b_{\phi(n)}]_n\}$ where the numbers $b_1, \ldots, b_{\phi(n)}$, satisfying $1 \leq b_1 < \cdots < b_{\phi(n)} < n$, are all relatively prime to n.

We claim that we can also describe $(\mathbb{Z}/n\mathbb{Z})^*$ as the set

$$M_a^* = \left\{[b_1 a]_n, \ldots, [b_{\phi(n)}]_n\right\} .$$

Note first that since each b_j is relatively prime to n, and so is a, it follows that $b_j a$ is relatively prime to n as well – the primes which divide $b_j a$ divide either b_j or a, and there are no such primes which also divide n. Therefore $M_a^* \subseteq (\mathbb{Z}/n\mathbb{Z})^*$.

Now notice that if $[b_j a]_n = [b_k a]_n$ for $1 \leq j, k \leq \phi(n)$ then $n \mid (b_j - b_k)a$. By Euclid's Lemma 2.1.6, since $\gcd(a, n) = 1$, we must have $n \mid (b_j - b_k)$.

However, since $1 \leq b_1 < \cdots < b_{\phi(n)} < n$, it follows that $-n + 1 < b_j - b_k < n - 1$, and the only multiple of n in that range is 0. Thus $b_j = b_k$ and we conclude that all of the elements in the above description of M_a^* are distinct. Since there are $\phi(n)$ such elements and M_a^* is a subset of $(\mathbb{Z}/n\mathbb{Z})^*$, which has only $\phi(n)$ elements, it must be that $M_a^* = (\mathbb{Z}/n\mathbb{Z})^*$.

As in the previous proof, we conclude that the products of all the elements in M_a^* and in $(\mathbb{Z}/n\mathbb{Z})^*$ must be the same:

$$b_1 a \cdot \cdots \cdot b_{\phi(n)} a \equiv b_1 \cdot \cdots \cdot b_{\phi(n)} \pmod{n} .$$

Regrouping terms again, we see

$$a^{\phi(n)} b_1 \cdot \cdots \cdot b_{\phi(n)} \equiv b_1 \cdot \cdots \cdot b_{\phi(n)} \pmod{n} .$$

Now the b's are all invertible mod n, so we can multiply by the inverses, one by one, until we are left with

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

which finishes Euler's Theorem. □

Exercises for §3.4.

EXERCISE 3.4.1. Let n, a, and $b_1, \ldots, b_{\phi(n)}$ be as in the proof of Euler's Theorem. Show that $b_1 \cdot \cdots \cdot b_{\phi(n)} \equiv \pm 1 \pmod{n}$.

CHAPTER 4

Cryptology

Here are some Greek roots:

kryptos, $\kappa\rho\upsilon\pi\tau o\varsigma$: secret, hidden
logos, $\lambda\acute{o}\gamma o\varsigma$: word, study, speech
graph, $\gamma\rho\acute{\alpha}\varphi\omega$: write, written

From these (and others), English gets the words

cryptosystem: a set of algorithms for protecting secrets
cryptography: work done to make cryptosystems
cryptanalysis: work done to circumvent the protections of cryptosystems
cryptology: the union of cryptography and cryptanalysis,
often abbreviated simply to *crypto*.

Beware that *cryptography* is widely (but inappropriately!) used as a synecdoche for *cryptology*. (This is not unlike the widely understood incorrect usage of the word *hacker*.) We will try to use these words more carefully.

With that understood, we start with a little elementary *cryptology* in this chapter. There will be very little number theory, but we will set up some terminology and simple examples of cryptography and the corresponding cryptanalysis, with an emphasis on the old, historic, systems which are no longer viable in the modern age. Later chapters will come around quickly to modern, number theoretic techniques in crypto.

4.1. Some Speculative History

Perhaps there was a form of deception that preceded language – certainly many a house pet has feigned innocence despite the clear evidence of involvement in stealing treats. And even apiologists may not know if some lazy bees make up a story about a long excursion to a new flower patch when their Queen demands an accounting.

But among *homo sapiens*, probably as soon as there was language, there was lying. Of course, when two humans are face to face, both parties have some control, such as: the listener can make an attempt to evaluate the trustworthiness of speaker, they can both form their own judgements of the other's identity and therefore choose what they wish to share with each other, and the words of the speaker pass directly from their lips to the

listener's ears without the possibility of change of meaning in flight (absent considerations of ambient noise and so on).

A great deal changed with the invention of writing more than 5000 years ago. Words frozen in physical form, and the ideas they represent, can be taken and shared with a wide range of parties other than those with whom the original author wanted to communicate. In addition, if an author is not able to hand her work directly to the intended reader and instead the written words are out in the world on their own for a while, then both intended communicants can no longer be sure that the other is who the writing claims them to be nor that the writing remains the unchanged symbols that the other party originally set down.

Let us formalize some of these issues of *information security* (as it is called now), in the context of a message to be sent from someone named *Alice* to someone named *Bob*. The role of the possibly disruptive and overly intrusive environment is played in our little drama by *Eve*. (Traditionally one skips directly to a character whose name starts with *E* to symbolize both the *environment* and also someone who is potentially an *eavesdropper* [1].)

Here is some basic terminology:

DEFINITION 4.1.1. **Confidentiality** means that only the intended recipient can extract the content of the message – Alice wants only Bob to get her message and not Eve, no matter if she listens on the eavesdrop or intercepts the message as it travels in some form from Alice to Bob.

DEFINITION 4.1.2. Message **integrity** means that the recipient can be sure the message was not altered – Bob wants to know that what he gets is what Alice wrote, not what the mischievous Eve intended to change it into.

DEFINITION 4.1.3. Sender **authentication** means that the recipient can determine from the message the identity of the sender – Bob wants to be sure this message did in fact originate with Alice.

DEFINITION 4.1.4. Sender **non-repudiation** means that the sender should not be able to deny sending that message – Bob wants to be able to hold Alice to her promises.

Note that Alice and Bob may actually be the same person sending a message from a past self to their future self. For example, someone may want to keep records which must be *confidential* and whose *integrity* must be reliable, even if there is a break-in to the site keeping those records.

One of the earliest schemes attempting to achieve confidentiality was probably used around the 7^{th} century BCE in Greece by military commanders who wanted to get dispatches from far away soldiers, and to send them orders, in a way that even if the messenger

[1] *Eavesdropper* apparently comes from the Old English *yfesdrype*, meaning literally *one who stands on the eavesdrop* [ground where water drips from the eaves of the roof] *to listen to conversations inside a house.*

is captured, the message they carry will not be readable by the enemy. They used a device called a **scytale** (the "c" is hard and the word rhymes with "Italy"), which was a cylindrical stick – maybe a spear shaft – ($\sigma\kappa\upsilon\tau\acute{\alpha}\eta$ means *baton* in ancient Greek), around which a long strip of parchment was wrapped in a spiral. A message could then be written in parallel rows *along the length of the scytale*, so that when the strip was unwound, the letters were all jumbled and the message was unreadable.

FIGURE 4.1.1. A scytale in use.[2]

On the other end, assuming the receiver also has a scytale, when the strip is wound in a spiral and read lengthwise, the message reappears, as if by magic.

Before we go on, a few more technical terms:

DEFINITION 4.1.5. The message that Alice wishes to send to Bob, in its actual original (and final) form is called the **plaintext**.

DEFINITION 4.1.6. An algorithm that Alice uses to transform (obfuscate) the plaintext into a form that will remain confidential even if observed by Eve while in flight is called a **cipher**. We also say that Alice **encrypts** the (plaintext) message.

DEFINITION 4.1.7. After the message has been encrypted, it is called the **ciphertext**.

DEFINITION 4.1.8. When Bob receives the ciphertext, he applies another algorithm to **decrypt** it and recover the plaintext.

[2]Image by DMGualtieri, CC-BY-SA-3.0 `http://creativecommons.org/licenses/by-sa/3.0`, via Wikimedia Commons, downloaded from `https://commons.wikimedia.org/wiki/File%3AScytale.png`

Graphically:

Basic crypto terminology:

Alice	on public network	Bob
plaintext/cleartext message m **encrypts** m to c transmits c	$\rightarrowtail$ **ciphertext** c $\rightarrowtail$	receives c **decrypts** c to recover plaintext m

Suppose a spy sees scouts in the field wrapping strips of parchment around their spears and writing down the length of the spear: meaning Eve learns the encryption algorithm. Although the *algorithm* is known, confidentiality may be preserved:

DEFINITION 4.1.9. Additional information (some of which is) used in encryption and (some of) which is necessary for successful decryption is called a **key**.

In the case of encryption with a scytale, the diameter of the scytale is the key. In fact, there is some conjecture that the scytale was also a simple from of authentication: only Alice had the matching scytale to the Bob's, so if Bob could read any coherent sequence of letters at all along his scytale, he knew Alice had sent it.

Even with a key, the vulnerabilities of this cryptosystem are fairly clear. And Athens did lose the Peloponnesian War.

Many hundreds of years of cryptology have shown that in fact it is better to reveal the algorithms of your cryptosystem to the public, and only to keep the key for a particularly communication channel (between Alice and Bob, say) secret. There are many reasons for this, the primary one probably being that the author of a cryptosystem can never be sure they have thought of all the possible methods of cryptanalysis which will be used against it. So the system's author is better off letting the general crypto community do its best against the system, and only the systems which have withstood such assault should be used. After all, this is basic to the scientific method itself: scientists publish all the details of their experiments, so others can try them as well and give independent verification or find something to criticize.

This idea – that the security of a cryptosystem should be based on the secrecy of the key but not of the algorithm – has come to be called **Kerckhoff's Principle**, after a set of cryptosystem design ideas written down in 1883 by a French military cryptography. It is held in opposition to cryptosystems, thought of as very weak, which rely upon no one ever finding out the algorithms: such systems rely instead on the ill-advised **security through obscurity** paradigm.

Exercises for §4.1.

EXERCISE 4.1.1. Suppose for some $k \in \mathbb{N}$, the cleartext Alice wishes to send consists of some symbols $m_1, \ldots, m_k$. Assume that the scytale she uses has a diameter such that $s \in \mathbb{N}$ letters can be written on each turn of the spiral of parchment, when it is wrapped around the scytale.

Write one or more formulæ describing what the letters $c_1, \ldots, c_k$ of the ciphertext will be. You should assume $s < k$ and, if you like, that they have any particular useful relationship such as $s \mid k$.

EXERCISE 4.1.2. Even if the scytale cryptosystem were strong, there may be problems using it for *authentication*. Describe how authentication might fail depending upon the message being sent – give an example or two of "bad" messages for authentication.

EXERCISE 4.1.3. The scytale cryptosystem seems weak. Describe how you would cryptanalyze it.

4.2. The Caesar Cipher and Its Variants

Another system which dates to ancient times was supposedly used by Julius Caesar.

DEFINITION 4.2.1. Alice takes her message, removes all spaces and punctuation, and puts it all in one case (maybe upper case). Then she moves each letter k places down the alphabet, wrapping around from Z to A if necessary, where $k \in \mathbb{Z}$ is a fixed number known to both Alice and Bob but no one else, called the **key**.

To decrypt, Bob simply moves each letter k places earlier in the alphabet, wrapping past A to Z if necessary: *i.e.* Bob encrypts the ciphertext with key $-k$ to get the plaintext.

This is called the **Caesar cryptosystem**.

Apparently Julius Caesar usually used the key value $k = 3$. His nephew Octavian, who later became the emperor Augustus, liked to use $k = -1$.

When using what is called the Latin alphabet (which is not what was used in ancient Rome, though), with its 26 letters, there is one particularly nice key value: 13. The nice thing about that value is encryption and decryption are exactly the same transformation. In modern times, this transformation is called **ROT13**, and it has a small role in the modern history of the Internet. In particular, posts on early chat rooms and bulletin boards would sometimes want to have a bit of content that should not be automatically available to anyone who looks at the post, but would be there for the determined reader (such as, for example, a spoiler in a review of some popular new game, book, or film). These were often included in the post, but only after they had been run through ROT13.

A few commercial products used ROT13 for actual security, despite it actually being completely insecure, such as certain parts of some versions of the Windows operating system.

The Caesar ciphers were completely broken (in a number of ways; see §4.3, below) before 1000CE, but a descendent was developed in the late Middle Ages and was considered the state of the cryptological art through the early modern period.

DEFINITION 4.2.2. Once again, Alice takes her message, removes all spaces and punctuation, and puts it all in one case (maybe upper case). This time, the key $\vec{k} = (k_1, \ldots, k_\ell)$ which Alice and Bob share is an ℓ-tuple, for $\ell \in \mathbb{N}$, of integers $k_1, \ldots, k_\ell \in \mathbb{Z}$.

To encrypt, Alice steps through her plaintext message and key sequence both one letter at a time, moving each plaintext letter forward (wrapping from A to Z if necessary) by the number of letters specified by the corresponding key number. If she runs out of key numbers, she simply starts again at the beginning of the key sequence.

To decrypt, Bob simply moves each letter k places earlier in the alphabet, wrapping past A to Z if necessary: *i.e.* to decrypt, Bob encrypts with key $-\vec{k} = (-k_1, \ldots, -k_\ell)$.

This is called the **Vigenère cryptosystem**.

Traditionally, the key in Vigenère is a written down, memorized, and shared between Alice and Bob in the form of a word. Then when encrypting or decrypting, the letters of the keyword are "added" to the message letters, as described above, under the convention that $A = 1$, $B = 2$, *etc.*.

Notice that Vigenère with key length ℓ is essentially ℓ Caesars in parallel – in fact, if $\ell = 1$, they are exactly the same cryptosystem. It therefore turns out that Vigenère is essentially $\ell + 1$ times harder to crack than Caesar, the "$+ 1$" coming from the fact that Eve doesn't even know ℓ. Nevertheless, after a couple of hundred years in which it was considered unbreakable and used as the principle diplomatic cipher in the courts of Europe, approaches to breaking Vigenère were developed.

As one final variant of the Vigenère cipher, suppose we go in the opposite direction from the $\ell = 1$ extreme, and instead take ℓ as long as possible.

DEFINITION 4.2.3. A **one-time pad** is a Vigenère cryptosystem in which the key is as long as the message, chosen randomly, and never re-used. [The key is also called the *one-time pad* in this cryptosystem[3].]

(A one-time pad is sometimes called – inappropriately, given the true intellectual history of the cryptosystem – a *Vernam Cipher* .)

It is important to have a good key sequence in a one-time pad cryptosystem. The good news is that one can prove that with a truly random one-time pad, the resulting cryptosystem is in fact perfectly secure. In computer science, this is called *information theoretically secure* , because the proof of security does not rely upon any assumptions about the computational resources available to the attacker. [See [**Sha49**] and [**Sha48**] for this proof.] Other cryptosystems we shall meet below are only secure if we assume the attacker has access to a computer of a particular type (a *probabilistic polynomial-time Turing machine* is the usual assumption; this is the subject within computer science called *computational complexity*, see, *e.g.,* [**AB09**]).

It is also important never to use the same pad more than once. If so, an attacker can take the letter-by-letter difference of the two ciphertexts and this will completely remove the pad from the calculation. In fact, at that point the message can usually be determined quickly.

In modern times, after the advent of digital communications networks, messages are written as computer data, so everything is stored (and transmitted) as bits, meaning 1s and 0s. With bits, when we do the equivalent of what was described above as shifting a letter along the alphabet, wrapping from Z to A if necessary, we are either doing nothing, if the shift is by an even amount, or simply switching 0 and 1 otherwise.

[3]Another synecdoche!

Another way of saying that is to say that is that if the message and the key are both written all out in bits – think of them as the elements $[0], [1] \in \mathbb{Z}/2\mathbb{Z}$ – then the encryption consists exactly of adding the corresponding bits mod 2. A good one-time pad is therefore a very random, and very long, string of bits (congruence classes in $\mathbb{Z}/2\mathbb{Z}$).

The big problem with one-time pads is *key distribution:* Alice and Bob must share very large one-time pads before they ever start to communicate, large enough to have as many letters (or bits, in the computer age) as all of their future messages.

This does suggest an interesting approach to cryptography: find a way that Alice and Bob can share a small secret out of which they can generate arbitrarily long sequences of bits, both getting the same sequence, which seem to any attacker to be entirely random. If this were possible, they would use these *pseudorandom* sequences (called this because they are not actually random – after all, they can be determined in advance by both Alice and Bob using their small starting secret – they only appear so to all attackers) as one-time pads. [See [**Lub96**] for more about pseudorandomness.]

Exercises for §4.2.

EXERCISE 4.2.1. ROT13 was supposedly nice because it reduced the amount of coding which needed to be done: the same program would decrypt as the one which encrypts, since doing ROT13 a second time undoes the first ROT13's transformation.

Is this true for other keyed Caesar ciphers? Make a conjecture about whether, given a Caesar cipher key $k \in \mathbb{Z}$ and a message m, there always exists an $n \in \mathbb{N}$ such that

$$\overbrace{e_k^C \circ \cdots \circ e_k^C}^{n \text{ times}}(m) = m$$

where e_k^C is the function which does the Caesar encryption with key k. If so, find an expression for the smallest such n, which depends (if necessary) on k, m, and the size of the alphabet in which m is written.

EXERCISE 4.2.2. Continuing the previous exercise: Suppose now $\vec{k} = (k_1, \ldots, k_\ell)$ is an ℓ-tuple, for $\ell \in \mathbb{N}$, of integers $k_1, \ldots, k_\ell \in \mathbb{Z}$ and $e_{\vec{k}}^V$ is the Vigenère encryption function with key $\vec{k}$. Find if for all messages m, there exists an $n \in \mathbb{N}$ such that

$$\overbrace{e_{\vec{k}}^V \circ \cdots \circ e_{\vec{k}}^V}^{n \text{ times}}(m) = m$$

and, if so, find an expression for the smallest such n, which depends (if necessary) on $\vec{k}$, m, and the size of the alphabet in which m is written.

EXERCISE 4.2.3. Suppose Eve has a twin Vev, and they are both looking at an intercepted ciphertext c sent by Alice to Bob. They think that those crazy lovebirds Alice and Bob have used a one-time pad to encrypt their communications, but Eve and Vev both do lots of calculations and think they have managed to correctly decrypt c. Unfortunately, they have different (non-gibberish!) values m_e and m_v which Eve and Vev, respectively, think were the plaintext that Alice meant to send to Bob.

What must have been the one-time pads p_e and p_v which they were somehow calculating (or guessing) that Alice used for their respective encryptions?

Could they have come up with arbitrary proposed decryptions m_e and m_v, or is there some relationship between the proposed decryptions, the proposed pads p_e and p_v, the true plaintext, and Alice's actual one-time pad?

Work out a particular, concrete example: if Alice's original cleartext message m was `aardvark`, would it be possible for Eve to think the message was $m_e =$ `iloveyou` while Vev thinks it is $m_v =$ `ihateyou`? If so, give an example of what would be the corresponding ciphertext c, Alice and Bob's one-time pad p_{AB}, the proposed decrypts m_e and m_v, and the proposed one-time pads p_e and p_v.

4.3. First Steps into Cryptanalysis: Frequency Analysis

The Caesar cipher seems very weak. But looking at a ciphertext, such as

```
wrehruqrwwrehwkdwlvwkhtxhvwlrq
zkhwkhuwlvqreohulqwkhplqgwrvxiihu
wkhvolqjvdqgduurzvrirxwudjhrxviruwxqh
ruwrwdnhdupvdjdlqvwdvhdriwurxeohv
dqgebrssrvlqjhqgwkhp
```

it is hard to know where to start – this hardly seems to be English at all.

Perhaps we should start with the Caesar cipher itself, assuming (anachronistically) that Caesar was following Kerckhoff's Principle, or that (more chronistically) spies had determined the cryptosystem but not the key.

One thing we notice right away is that what we do not know, that key, is such a small thing to be giving us such trouble. In fact, if we tried a random choice of key, nearly four times out of 100 (using the modern "Latin alphabet" of 26 letters – in Caesar's time, the actual Latin alphabet had only 23, so even better!) would, purely by dumb luck, give us a correct decryption. Formally, we are talking about

DEFINITION 4.3.1. The set of all valid keys for some cryptosystem is called its **keyspace**.

EXAMPLE 4.3.2. The keyspace of the Caesar cipher cryptosystem is the alphabet.

EXAMPLE 4.3.3. The keyspace of the Vigenère cipher cryptosystem with a key length of ℓ is all sequences of ℓ letters in the alphabet; therefore, it has N^ℓ elements, if the alphabet is of size N. If the precise key length is not known, but it is known to be less than or equal to some bound $L \in \mathbb{N}$, then there are $\sum_{j=1}^{L} N^j$ possible keys.

EXAMPLE 4.3.4. Any sequence of letters of length M is a possible key (pad) for a one-time pad encryption of a message of length M. Therefore there are N^M possible one-time pads for a message of length M over an alphabet of size N.

We computed the sizes of these various keyspaces because one thing that seemed weak about the Caesar cipher was how small was its keyspace. In fact, a better strategy than simply guessing at random would be to try all possible keys – there are only 26 (or 23 in Julius's time)! – and see which one gives the correct decryption. Such an approach is called a *brute-force attack* [or *exhaustive search*]. Even in Caesar's time, the Caesar cipher keyspace is so small that Eve could check all possible keys and see which yielded the cleartext of a message from Alice to Bob.

Things are a little more difficult for the Vigenère cipher. For example, there are nearly 12 million keys to try if the key length is known to be five. Before the computer age, this would have been completely intractable. Even in the 21^{st} century, where it would be nearly instantaneous to human eyes for a computer to generate all of those possible decryption,

there is a problem: those human eyes would have to look over the 12 million possibilities and pick out which was the valid decryption.

Amazon's *Mechanical Turk* (see `https://www.mturk.com/mturk/welcome`) might be able to marshal enough human eyes to solve a single brute force Vigenère decryption. But a better approach would be to write a computer program that can distinguish gibberish from a real message Alice might have sent to Bob – then, in a matter of moments, a brute force attack would succeed.

If we are to distinguish gibberish from a real message, we need to know what possible real messages are. This motivates the following

DEFINITION 4.3.5. The set of possible messages for an encrypted communication is called the **message space**.

Without some structure for the message space, cryptanalysis can become nearly impossible. For example, if Alice is e-mailing an encrypted version the combination of a safe to Bob, the message space is not too large but has no identifiable features: a brute force attack would have no way of telling which potential decryption was the actual combination.

But suppose we assume that the message space of Alice and Bob's communication is a set of short (or long, which would be better) English texts. There is quite a bit of structure in English texts – we human speakers of English certainly recognize valid English. The question is whether we can automate the process of detection of valid English texts.

Looking back at the example Caesar encryption given at the beginning of this section, we noticed that it did not look like English – but in what way? One thing was, of course, that it does not have any English words in it. Someone who knows English knows most of the words of that language and can recognize that this text has none of them, except maybe "up" and "i" (several times).

This suggests an approach: take a list of all words in English, with all variant and derivative forms, add in some possible short random sequences that may have been included in otherwise standard English (where Alice was quoting some nonsense she'd heard, or was writing down a graffito she saw on the side of a subway car, *etc.*), and compare them to possible decryptions of the ciphertext. This scheme is not very practical. But if the message space were small enough, something like this would be reasonable.

Looking back at the message again, another thing that immediately strikes the eye of a reader of English is that it does not seem to have the right letters, or the right combinations of letters. For example, there do not seem to be enough vowels. Looking at how often letters occur in a text compared to what we expect is called *frequency analysis*. Its use in cryptanalysis was apparently first described by the Muslim philosopher and mathematician al-Kindi[4] in his 9^{th} century work *Manuscript on Deciphering Cryptographic Messages*.

[4]Al-Kindi is a very interesting figure from early Muslim intellectual history: *e.g.,* he seems to have brought Hindu numbers, with their place-value notation, into the Muslim world.

Here is a table of the frequencies of the letters in a bit of standard English (some of the plays of Shakespeare):

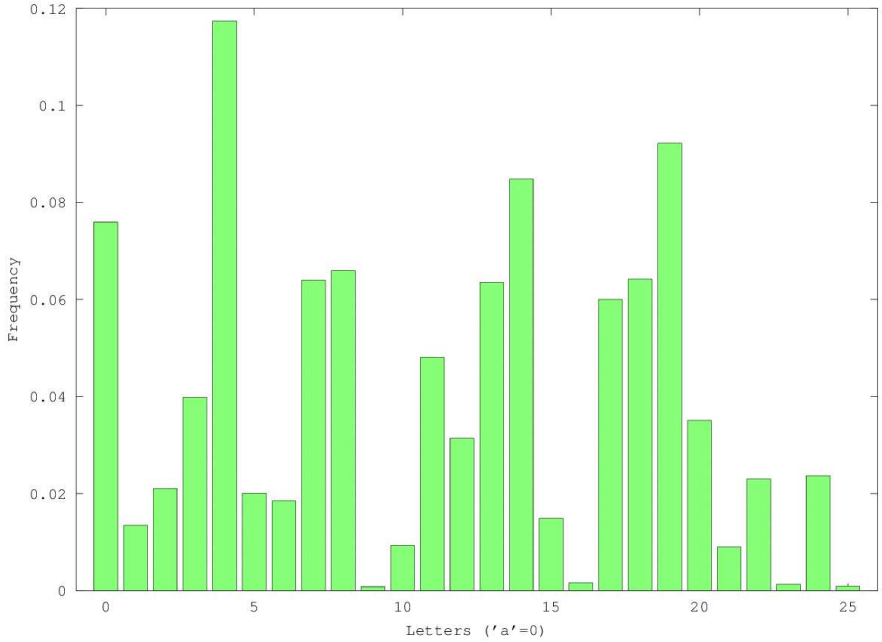

FIGURE 4.3.1. Frequency counts of letters in several Shakespeare plays

Notice that the letter ‘e’ is the most common, by a bit. So a first cryptanalytic approach using frequency analysis would simply be to use the Caesar decryption key which moved the most frequent letter in the ciphertext to be ‘e’. With the ciphertext at the beginning of this section, that would yield the decryption:

```
ezmpzcyzeezmpesletdespbfpdetzy
hspespcetdyzmwpctyespxtyoezdfqqpc
espdwtyrdlyolcczhdzqzfeclrpzfdqzcefyp
zcezelvplcxdlrltydeldplzqeczfmwpd
lyomjzaazdtyrpyoespx
```

Not so good. Apparently looking only at the most frequent letter is insufficient: the most frequent letter in the plaintext of this message was not ’e’.

Instead let us look at the entire frequency table for this ciphertext.

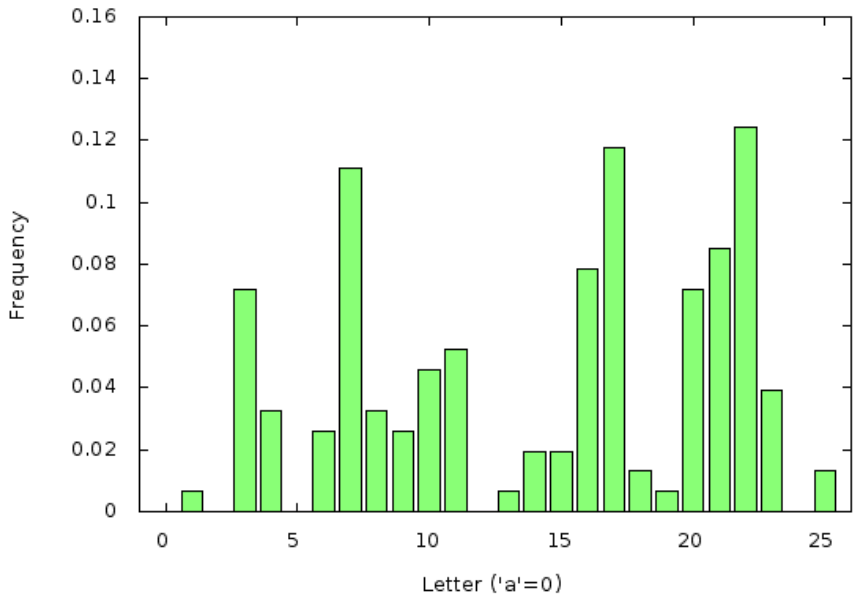

FIGURE 4.3.2. Letter frequencies in a sample Caesar ciphertext

Unfortunately, this is not exactly the standard (Shakespearean) English frequency distribution, nor is it even a shifted (with wrapping past 25 back to 0) version of Shakespeare's distribution. But perhaps one of its shifts is fairly close to Shakespeare.

This suggest the following more refined cryptanalytic approach: try all possible decryptions (which is not so bad for the Caesar cipher, because of its small keyspace) and choose the one whose entire letter frequency table is closest to the standard English letter frequency table – so, *not* looking only at the most frequent letter, but looking at the whole frequency distribution; and *not* looking for exact match, but merely for the best approximate match.

Which brings up the question of what is the best way to measure the distance between two distributions. One commonly used approach is to measure the

DEFINITION 4.3.6. The **total square error** is a distance between two distributions f and g given by

$$d(f,g) = \sum_{j=0}^{25} (f(j) - g(j))^2$$

where we are always working with our distributions defined on the letters in the form 'a'=0, 'b'=1, *etc.*).

Minimizing this distance is equivalent to *least squares* in statistics or linear algebra.

Following this strategy with the ciphertext from the beginning of this section, we get the following

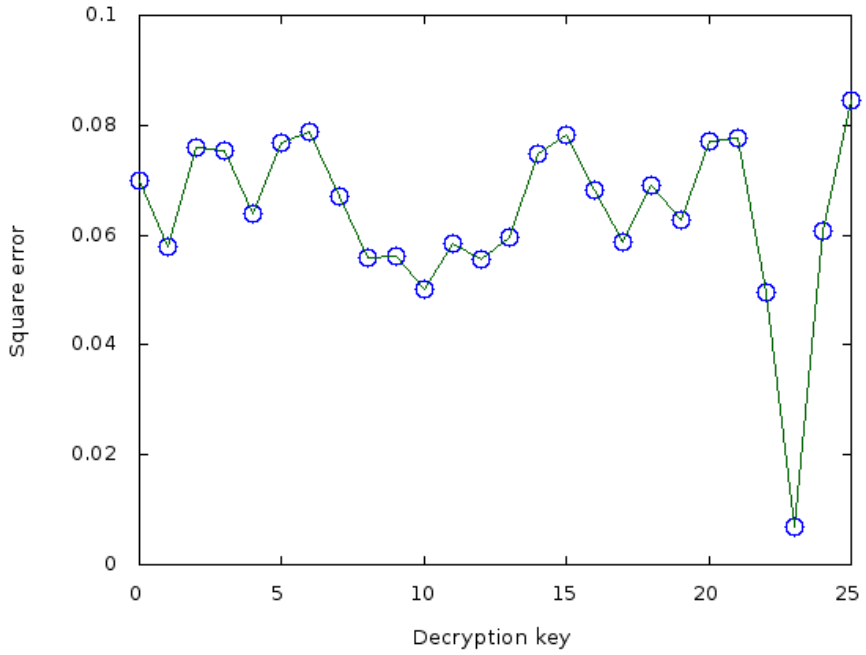

FIGURE 4.3.3. Square errors with all decryption keys for example Caesar ciphertext

The minimal square error will then come from decryption key 23 (corresponding to an original encryption key of 3; this was done by Julius himself, anachronistically). The corresponding cleartext then would have been

```
tobeornottobethatisthequestion
whethertisnoblerinthemindtosuffer
theslingsandarrowsofoutrageousfortune
ortotakearmsagainstaseaoftroubles
andbyopposingendthem
```

which looks pretty good.

Interestingly, this approach to non-gibberish detection is fairly robust, even with quite short texts which might be expected to have frequency distributions rather far away from the English standard. For example, with ciphertext

```
rkkrtbrkeffespkyvtifjjifruj
```

we have

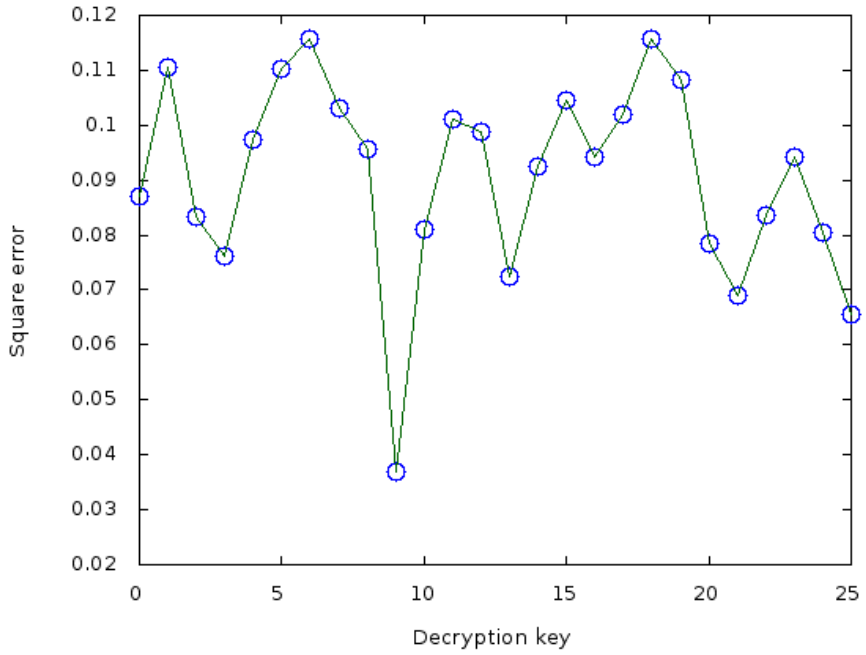

FIGURE 4.3.4. Square errors with all decryption keys for Caesar ciphertext `rkkrtbrkeffespkyvtifjjifruj`

from which we conclude that the decryption key is 9, corresponding to an encryption key of 17. Thus the cleartext must have been

`attackatnoonbythecrossroads`

which seems like the kind of thing Julius might have said.

So much for cryptanalysis of the Caesar cipher. Moving on to Vigenère, the first problem we face is finding out the key length. Because if we knew that key length ℓ, we would divide the ciphertext into ℓ shorter texts consisting of every ℓ^{th} character starting with the first, then with the second, *etc.,* up to starting with the ℓ^{th}. Applying the automatic Caesar cracker described above, we would have the ℓ digits of the key and thus the plaintext.

To make things easier, suppose we take all of *Hamlet* and encrypt it with Vigenère using the keyword `hippopotomonstrosesquipedaliophobia`, corresponding to numerical key

$$k = (7, 8, 15, 15, 14, 15, 14, 19, 14, 12, 14, 13, 18, 19, 17, 14,$$
$$18, 4, 18, 16, 20, 8, 15, 4, 3, 0, 11, 8, 14, 15, 7, 14, 1, 8, 0)$$

For example, the part of the ciphertext corresponding to the quote used above in Caesar cracking would be

```
hdxajnioomjvdtfvstrqitmfozlxjj
kcmiosgpenjjbgxmcmtfzltmaudofpmwg
odsntxuuhwjywmrjpnieosoqlfbpjoqyyljia
cmbdaozawminabtdhrtyndlncugkflswh
vjrwgdwddoeicznymcyl
```

If Eve was hoping this was only Caesar-encrypted, so the Vigenère key length was $\ell = 1$, she would have the following letter frequencies:

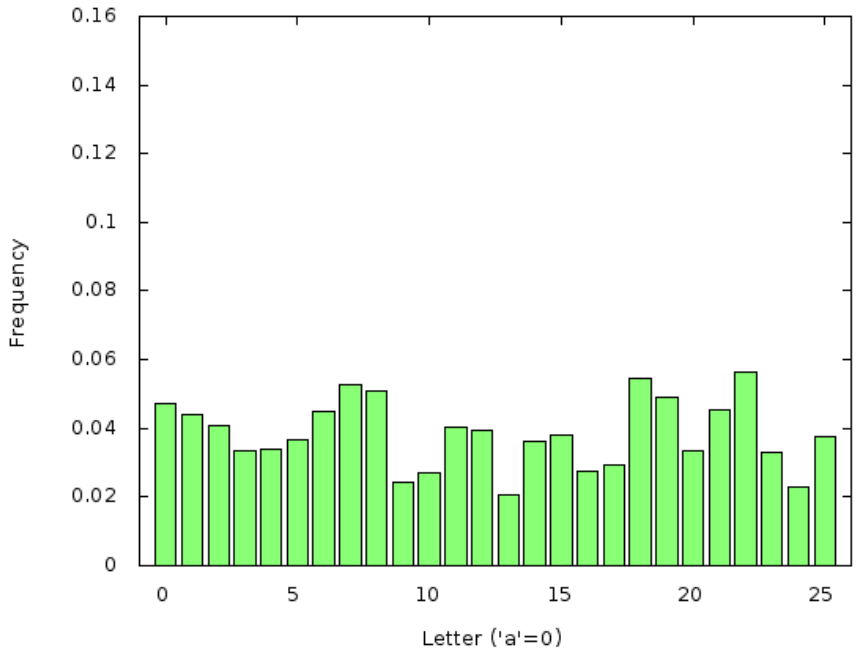

FIGURE 4.3.5. Letter frequencies in the Vigenère encryption of *Hamlet* using key `hippopotomonstrosesquipedaliophobia`

Notice that compared to Figure 4.3.2, the values show much less variation – in fact, there are no zero values and no values which are much larger than all others. The reason is that this distribution is the average of 16 different shifts, coming from the 16 different letters in the keyword `hippopotomonstrosesquipedaliophobia`, of the standard English distribution Figure 4.3.1. This averaging smooths out all of the characteristic shape of the distribution useful in identifying when its decrypting shift is correct.

Nevertheless, the Caesar cracker will give some key value for each one of the choices Eve makes for a possible value of ℓ – there will always be a value of the shift for each letter which minimizes the square error. The minimum will not be very small, though, because the frequency distribution for the letters chosen as all locations in the ciphertext congruent to $k \pmod{\ell}$ for any $k \in \mathbb{N}$ and $k \leq \ell$, where ℓ is less than the true key size, will be quite flat and therefore far away (in the square error distance) from the distribution of standard English.

For example, with the encrypted *Hamlet* we have been examining, if Eve guesses that $\ell = 5$ and looks at every letter in a location congruent to $1 \pmod 5$, the Caesar cracker will have the following graph of square error:

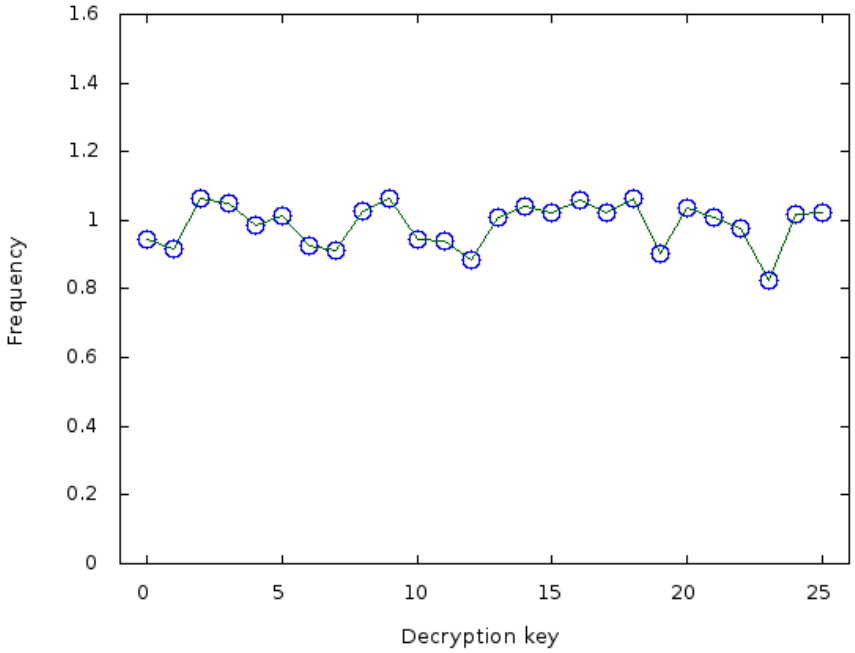

FIGURE 4.3.6. Square errors with all decryption keys for letters at locations congruent to $1 \pmod 5$ in *Hamlet* Vigenère encrypted with key `hippopotomonstrosesquipedaliophobia`

The minimum is clearly at a decryption key of 23, but it is not significantly less than other possibilities – and all the errors are around ten times higher than we saw in earlier such graphs such as Figure 4.3.3.

So here is the Vigenère cryptanalytic approach: Try all key lengths ℓ up to some bound L (determined by how much computer time is available, and not so large that taking every ℓ^{th} letter of the ciphertext results in too few letters to do reasonable frequency analysis). For each of the ℓ substrings consisting of every ℓ^{th} letter starting at locations $1, \ldots, \ell$, apply the Caesar cracker. This will result in an optimal Vigenère key $\vec{k}_\ell = (k_{\ell,1}, \ldots, k_{\ell,\ell})$ of length ℓ. Then for each such $\vec{k}_\ell$ in the range $1, \ldots, L$, compute the square error distance d_ℓ between the ciphertext decrypted with $\vec{k}_\ell$ and the standard English letter distribution. Report ℓ and $\vec{k}_\ell$ as the Vigenère keylength and key if d_ℓ is the smallest square error computed.

Exercises for §4.3.

EXERCISE 4.3.1. Can you think of any Caesar ciphertexts which would have two decryptions that would both seem like valid English during an brute-force attack? If so, give precise examples, with decryption keys.

EXERCISE 4.3.2. Would it be harder or easier to do the previous exercise (4.3.1) for the Vigenère cipher? Give reasoning and examples.

EXERCISE 4.3.3. It seems like more encryption would be better, in terms of designing a secure cryptosystem – so how about encrypting a cleartext using one cryptosystem, then re-encrypting the resulting ciphertext to make a super-ciphertext?

We have two cryptosystems so far with short keys (we also have the one-time pad, but it is already perfectly secure and has big keys, those pads, so we will not consider it): Caesar and Vigenère. Suppose I make a new cryptosystem which does some combination of Caesar and Vigenère encryptions, one after the other, all with different keys. Will this result in a substantially more secure cryptosystem? Why or why not?

4.4. Public-Key Crypto: the RSA Cryptosystem

Suppose Alice and Bob never had a chance to meet in person, and they nevertheless want to exchange messages which will be secret from Eve. What can they do?

This seems impossible within the context of cryptosystems we have been discussing so far. Let us tease out the crucial part that makes this so difficult

DEFINITION 4.4.1. A **symmetric cipher** (or **symmetric cryptosystem**) consists of the following parts, all known to both the communicating parties and the public in general:

- a message space $\mathcal{M}$
- a keyspace $\mathcal{K}$
- an encryption algorithm which creates a ciphertext $c = e_k(m)$ for any choice of $k \in \mathcal{K}$ and $m \in \mathcal{M}$.
- a decryption algorithm which generates a message $d_k(c) \in \mathcal{M}$ given a ciphertext c and a $k \in \mathcal{K}$, which satisfies $d_k(e_k(m)) = m \; \forall k \in \mathcal{K}, m \in \mathcal{M}$.

Alice and Bob can use such a symmetric cipher by agreeing in private upon a key $k \in \mathcal{K}$ they will use for both encryption and decryption. For this reason, symmetric cryptosystems are also called **private key cryptosystems**.

Graphically:

Basic private key crypto set-up and notation:

Alice $\longleftrightarrow$	private communication of shared key $k \in \mathcal{K}$ $\longleftrightarrow$	Bob
	on public network	
message $m_A \in \mathcal{M}$ compute $c_A = e_k(m_A)$ transmit c_A	$\rightarrowtail$ ciphertext c_A $\rightarrowtail$	receive c_A compute $m_A = d_k(c_A)$
receive c_B compute $m_B = d_k(c_B)$	$\leftarrowtail$ ciphertext c_B $\leftarrowtail$	message $m_B \in \mathcal{M}$ compute $c_B = e_k(m_B)$ transmit c_B
etc....		

It is important for the security of the above cryptosystem that the keyspace $\mathcal{K}$ either be quite large, or that the whole system be structured in such a way that it is hard to tell if a particular possible decryption is valid or not (or both). Otherwise Eve can perform the brute-force attack of simply trying all the possible keys $k \in \mathcal{K}$ and seeing which $d_k(c)$ makes sense as the decryption.

In any case, because of the initial private key exchange, symmetric cryptosystems are not appropriate for the kind of situation we proposed at the start of this section. Instead, one would need a different kind of cryptosystem:

DEFINITION 4.4.2. An **asymmetric cipher** (or **asymmetric cryptosystem**) consists of the following parts, all known to any interested party, licit or il-:

- a message space $\mathcal{M}$
- an encryption keyspace $\mathcal{K}_e$
- a decryption keyspace $\mathcal{K}_d$
- a way of associating encryption keys to decryption keys, by a one-to-one function $\mathcal{E} : \mathcal{K}_d \to \mathcal{K}_e$
- an encryption algorithm which creates a ciphertext $c = e_{k_e}(m)$ for any choice of $k_e \in \mathcal{K}_e$ and $m \in \mathcal{M}$.
- a decryption algorithm which generates a message $d_{k_d}(c) \in \mathcal{M}$ given a ciphertext c and a $k_d \in \mathcal{K}_d$, which satisfies $d_{k_d}(e_{\mathcal{E}(k_d)}(m)) = m \; \forall k_d \in \mathcal{K}_d, m \in \mathcal{M}$.

To use such a cryptosystem, Bob picks a decryption key k_d, without telling it to anyone – but he makes the corresponding encryption key $k_e = \mathcal{E}(k_d)$ publicly available (probably on his website). For this reason, k_e is also called his **public key** and k_d his **private key**, while the whole system is called a **public-key cryptosystem**. .

Graphically:

Basic public key crypto set-up and notation:

Alice	on public network	Bob
		pick $k_d \in \mathcal{K}_d$ compute $k_e = \mathcal{E}(k_d) \in \mathcal{K}_e$
download k_e	$\longleftarrow$ public key k_e $\longleftarrow$	publish k_e
message $m \in \mathcal{M}$ compute $c = e_{k_e}(m)$ transmit c	$\longrightarrow$ ciphertext c $\longrightarrow$	receive c compute $m = d_{k_d}(c)$

As before, the security of this cryptosystem against brute-force attacks relies upon the size of $\mathcal{K}_d$. In addition, since the encryption key and algorithm are known to Eve, it is important that the message space $\mathcal{M}$ not be too small. If $\mathcal{M}$ were small, Eve could simply compute all encryptions $e_{k_e}(m)$ as m ran over $\mathcal{M}$, and compare them to a ciphertext c she intercepted.

There is a way to prevent this attack on the message space, by making it artificially larger. In fact, one usually adds some

DEFINITION 4.4.3. **Cryptographic salt** is random data added to a message Alice desires to send to Bob before she encrypts it, which is automatically removed at Bob's end.

Once Alice salts her messages, Eve must search the entire space of messages together with salt, which can be much larger than $\mathcal{M}$.

As an added benefit, salting messages makes the ciphertext change with each secret transmission – even if the plaintext is the same. Therefore even though Eve will of course know there is communication going on between Alice and Bob, her traffic analysis will no longer even be able to track when the same message is being sent on separate occasions. Absent this salt, a meticulous Eve might be able to correlate the message ciphertext with actions in the real world, and thus effectively learn their decryption even without cracking the cryptosystem. With salt, this is impossible.

Since Bob's encryption key $k_e = \mathcal{D}(k_d)$ is public, the entire security of this cryptosystem falls apart if Eve can figure out the inverse of the function $\mathcal{E} : \mathcal{K}_d \to \mathcal{K}_e$. On the other hand, Bob must have been able to compute $\mathcal{E}$ itself, at least during the set-up phase of the cryptosystem. There is a name for this kind of function

DEFINITION 4.4.4. A function $f : A \to B$ which is one-to-one and which can be computed in a reasonable amount of time on a standard (classical) computer but whose inverse cannot feasibly be computed is called a [**cryptographic**] **one-way function**.

Multiplying two numbers, even very large numbers, can be done quite quickly on a computer. It is even surprisingly fast to tell if very large numbers are prime or composite (see [**AKS04**] for this amazing recent result in the long history of primality testing).

But no one has found a way to *factor* large numbers, even when they are known to be composite, in a reasonable amount of time.[5] For example, one of the largest ever to be factored, a famous challenge problem known as **RSA-768**, consisting of a 232-digit (768 bit) composite number, was finally factored in 2009 after a network of hundreds of computers working together (although not exclusively on this problem) had been processing for about two years. Larger composite numbers are easy to make and astronomically harder to factor.

Therefore, the function which multiplies two large primes together to get a very large composite is a good candidate for a cryptographic one-way function.

In 1978, Ron Rivest, Adi Shamir and Leonard Adleman described ([**RSA78**]) the following public-key cryptosystem based on this one-way function:

DEFINITION 4.4.5. Bob starts by picking two very large primes p and q. Their product $n = pq$ is known as the corresponding **RSA modulus**. Bob also picks a number e, typically a number which when written in base two has very few 1's, such as 3 or

[5] On a *classical* computer – there is an efficient algorithm known for factoring on a *quantum computer*, see [**Sho94**] and [**NC10**].

$65537 = 10000000000000001_2$, which satisfies $\gcd(e, \phi(n)) = 1$. This number is called the **RSA exponent**.

The **RSA public [encryption] key** k_e consists of the pair (n, e); the set of possible such pairs is $\mathcal{K}_e$.

The **RSA private [decryption] key** k_d consists of the pair (n, d), where

$$d = e^{-1} \pmod{\phi(n)},$$

and $\mathcal{K}_d$ is the set of such pairs.

The association of encryption to decryption keys is by $\mathcal{E} : (n, e) \mapsto (n, e^{-1} \pmod{\phi(n)})$.

The message space of this cryptosystem is $\mathcal{M} = \{m \in \mathbb{Z} \mid 0 \leq m < n\}$.

Encryption is by

$$c = e_{(n,e)}(m) = m^e \pmod{n}\ .$$

Decryption is

$$d_{(n,d)}(c) = c^d \pmod{n}\ .$$

All together, the above is called the **RSA cryptosystem**.

Graphically:

RSA cryptosystem set-up and notation:

Alice	on public network	Bob
download k_e	$\leftarrow\!\!\!\!\!\!\leftarrow$ public key k_e $\leftarrow\!\!\!\!\!\!\leftarrow$	pick large primes p and q compute **RSA modulus** $n = pq$ pick **RSA exponent** $e \in \mathbb{N}$ with $\gcd(e, \phi(n))$ publish $k_e = (n, e)$ compute $d = e^{-1} \pmod{\phi(n)}$
message $m \in \mathcal{M}$ compute $c = e_{(n,e)}(m)$ $= m^e \pmod{n}$ transmit c	$\rightarrow\!\!\!\!\!\!\rightarrow$ ciphertext c $\rightarrow\!\!\!\!\!\!\rightarrow$	receive c compute $m = d_{(n,d)}(c)$ $= c^d \pmod{n}$

The first thing we need to see is that this satisfies the basic condition to be a functioning cryptosystem:

PROPOSITION 4.4.6. *Let notation be as in the above definition of the RSA cryptosystem. Then for any message $m \in \mathcal{M}$, $d_{(n,d)}(e_{(n,e)}(m)) = m$.*

PROOF. Any $m \in \mathcal{M}$ represents a class in $\mathbb{Z}/n\mathbb{Z}$, and we will blur the distinction between the class and its representative m satisfying $0 \leq m < n$.

We have built d as $d = e^{-1} \pmod{\phi(n)}$. This means $e \cdot d \equiv 1 \pmod{\phi(n)}$, so $\exists k \in \mathbb{Z}$ such that $e \cdot d = 1 + k\phi(n)$.

Then by Euler's Theorem 3.3.4, $\forall m \in \mathcal{M}$ such that $\gcd(m, n) = 1$

$$d_{(n,d)}(e_{(n,e)}(m)) \equiv (m^e)^d = m^{ed} = m^{1+k\phi(n)} \equiv m \cdot (m^{\phi(n)})^k \equiv m \cdot (1)^k \equiv m \pmod{n}$$

as desired. The case where $\gcd(m, n) \neq 1$ is Exercise 4.4.1 in this section. □

Beyond basic functionality, we need to see how practical RSA is. So let us go through it carefully, picking out the algorithms needed at each step.

(1) Bob must choose the large primes p and q. As we have said, there are algorithms which can tell if a number is prime in a reasonable amount of time. More precisely, this means that a (classical) computer can determine if a number k is prime in an amount of time which is less than a polynomial function of $\log(k)$. [That's the usual meaning of *feasible computation* in cryptology.]

(2) Furthermore, there are enough primes that Eve will not be able to do a brute force search through all the possibilities. We can tell this because The Prime Number Theorem tells us the (asymptotic) number of primes:

THEOREM 4.4.7. *Given $x \in \mathbb{R}$, $x > 0$, let $\pi(x) = \#\{p \in \mathbb{N} | p \leq n\}$ be the* **prime counting function**. *Then*

$$\lim_{x \to \infty} \frac{\pi(x)}{x/\ln(x)} = 1.$$

This theorem is one of the gems of analytic number theory. It was proven in 1896 by Jacques Hadamard and Charles Jean de la Vallée-Poussin, although important pieces were known before then by people like Euler, Legendre, and Riemann. The proof is hard, even an *elementary* one from 1948 by Atle Selberg and Paul Erdös.

One consequence of the Prime Number Theorem is that the probability that a randomly chosen number k is prime is approximately $1/\ln(k)$. For example, looking for a prime with 200 digits, we pick a random number of that size, and then we will have to test on the order of $\ln(10^{200}) = 461$ numbers before we have a prime. This is not an unreasonable amount of computation.

(3) Having an RSA modulus $n = pq$, we can compute the Euler totient function $\phi(n) = (p-1)(q-1)$ nearly instantly by Theorem 2.5.3.

(4) Finding the RSA exponent e which is relatively prime to $\phi(n)$ is not hard, since there are many such – in fact, in a real sense, the probability that two randomly chosen numbers are relatively prime is $\frac{6}{\pi^2}$ (for a precise statement of this, and its proof, see [**HW79**]). Testing candidates for e is also efficient because, as we shall see below in Exercise 4.4.3, the Euclidean Algorithm takes an amount of time which is less than a polynomial function of the $\log$'s of the numbers involved – it is *feasible*.

(5) Next, computing $d = e^{-1} \pmod{\phi(n)}$ is very fast using the Extended Euclidean Algorithm (Theorem 1.6.2) which is certainly slower than the basic Euclidean Algorithm, but is still feasible.

(6) RSA encryption and decryption both consist of raising a number to a power in $\pmod{n}$. Fortunately, there is an algorithm called **fast modular exponentiation** which again does this in an amount of time less than a polynomial function of the $\log$'s of the given numbers.

(7) One last practical point is how Alice actually uses RSA to send the message she wants, and not merely a number $m \in \mathbb{Z}$ such that $0 \leq m < n$.

This is a standard problem in mathematical cryptography and has standard solutions. Typically, Alice must take her message, character by character, and transcribe it into a sequence of numbers using some accepted standard. This has been done since 1960 using the **ASCII code** [American Standard Code for Information Interchange], which gives 7 bit binary numbers for the 26 letters of the (modern) Latin alphabet, both upper and lower case, as well as for a set of commonly found symbols and a few modern additions, usually no-printing, such as 11110_2 *Record Separator*, 111_2 *Bell*, 1011_2 *Vertical Tab*, *etc.* These ASCII codes are today always stored in one byte (8 bits), padded with a leading 0 bit.

Actually, as the World Wide Web has become a global phenomenon, there has been an increasing need for more alphabets and even writing systems (like Chinese) which do not use alphabets. This has resulted in a system called **Unicode** which as, in version 6.3 as of 2013, more than 110,000 characters. Unicode is stored in a variety encodings, of which the most common are **UTF-8** and **UTF-16**, which use between two and four bytes to store each character.

What Alice typically does, then, is to write out her message in either ASCII or Unicode, simply attach all the bytes in sequence, and then cut the entire message into blocks of b bits. Here $b \in \mathbb{N}$ is chosen to be the largest value such that $2^b < n$, so that those blocks of b bits can be thought of as representing an integer $m \in \mathbb{Z}$ such that $0 \leq m < n$, so RSA can now be used on the message, a block at a time.

(We are glossing over many details here, such as how to leave space for some salt, other approaches and issues in the structure of the blocks, *etc.* See any standard reference on cryptography, such as [**FS03**] or [**MVOV96**].)

Exercises for §4.4.

EXERCISE 4.4.1. In this problem, you will finish the proof of Proposition 4.4.6, which states that RSA decryption functions correctly in all cases.

As a first step, formally state and prove a lemma that for distinct primes p and q, two integers r and s are congruent mod pq if and only if they are congruent mod p and mod q. *[Hint: the Chinese Remainder Theorem.]*

Now, using the above, prove the missing case of Proposition 4.4.6: Given distinct primes p and q, define $n = pq$. Let $e \in \mathbb{N}$ satisfy $\gcd(e, \phi(n)) = 1$ and $d \in \mathbb{N}$ be an inverse of e mod $\phi(n)$. Then, for any $m \in \mathbb{Z}$ such that $0 \leq m < n$ and $\gcd(m, n) \neq 1$, we have $m^{ed} \equiv m \pmod{n}$.

EXERCISE 4.4.2. How much computation is required to compute $a^k \pmod{n}$ for $a \in \mathbb{Z}$ and $k, n \in \mathbb{N}$ with $n \geq 2$?

Multiplying two numbers $a, b \in \mathbb{Z}$ is a basic instruction in most modern computers: it can be thought of as taking one unit of time. (Or you can do a number of smaller manipulations which is approximately a polynomial function of the number of digits in a and b; it will make no difference for the rest of this problem.) Similarly, division with remainder is either a single machine instruction (*e.g.* on a processor in the Pentium family) or can be done by elementary school methods in time approximated by a polynomial function of the sizes of the inputs.

Let us then describe the work to do a multiplication followed by reducing $\pmod{n}$ as being bounded by a polynomial function $p(d)$, where d is the number of digits in the operands.

If we then simply multiply a by itself k times, reducing $\pmod{n}$ each time, the time to do this will be approximately $kp(d) = c_1 e^{c_2 d} p(d)$, for some constants $c_1, c_2 \in \mathbb{R}$ – it will increase *exponentially*.

See if you can come up with a much fast exponentiation algorithm, polynomial rather than exponential.

[Hints: (1) Make a table of successive squares of a in $\pmod{n}$, being a^2, a^4, a^6, etc. (2) write out k in base two, and expand a^k using this binary version of k, the usual rules for exponentiation, and the table. End up with a description of how to compute $a^k \pmod{n}$, and check carefully how much time it would take to do your computation.]

EXERCISE 4.4.3. For $k \in \mathbb{N}$, play the following game:

(1) use the division algorithm on k for division by two, getting quotient q and remainder r;
(2) replace $k \leftarrow q$;
(3) if $k > 0$, repeat from step (1); otherwise, terminate.

Give a bound on how many steps does it take for this game to finish – answer in terms of a function of k, or a function of the number of bits required to write k down in base two, or a function of the number of digits require to write k in base 10.

Show that for every two steps of the Euclidean Algorithm, the remainder terms r_i decrease by a least a faction of $1/2$. In other words, if we use the notation of Theorem 1.6.2 then $r_{j+2} < \frac{1}{2} r_j$ for all $j \in \mathbb{Z}$ satisfying $j \geq 0$.

Explain how this means the Euclidean Algorithm requires at most $\log_2(b)$ steps to compute $\gcd(a, b)$ and it therefore requires at most seven times the number of digits of b. *[Hint: what is* $\log_2(10)$*?]*

Explain how this means that the Euclidean Algorithm is a cryptologically *feasible* computation, in the sense of this section.

4.5. Digital Signatures

Public-key cryptosystems allow several use-cases which symmetric cryptosystems do not. One which has come to have more and more importance in the modern digital economy is the creation of *digital signatures* – these are parts of electronic documents which are supposed to have something of the qualities of a physical signature in that are hard for an imposter to forge. Such a signed document might be needed, for example, if Bob from the last section (whose RSA public key is on his website) wished to send a legally binding contract via e-mail. Perhaps Alice and Bob wish to e-mail to their future landlord Larry a signed lease for an apartment that they will share. When Larry gets an e-mail from Bob saying "I agree to be bound by the terms of this lease," Larry needs to have confidence that this e-mail did originate from Bob, which he can if there is a digital signature.

Here's what Bob can do: he takes a copy of the lease, adds a section at the end stating his agreement to its terms and giving some personally identifying information (perhaps a scan of his driver's license). Call this whole chunk of data m. Then Bob applies his *decryption* algorithm, using his private (decryption) key k_d, yielding $s = d_{k_d}(m)$ – this s is called Bob's signature on the message m. He then e-mails both m and s to Larry.

When Larry receives this signed message, the first thing he does is detach the signature s and compute its encryption $e_{k_e}(s)$ using the public key he got off Bob's website. Since e_{k_e} and d_{k_d} are inverses and it does not matter in which order they are applied, the result should be m. If that is so, Larry can be sure that whoever sent the message also had access to Bob's secret key and so presumably is Bob himself.

Graphically:

Basic digital signatures:

Larry	on public network	Bob
		pick $k_d \in \mathcal{K}_d$ compute $k_e = \mathcal{E}(k_d)$
download k_e	$\longleftarrow$ public key k_e $\longleftarrow$	publish k_e
		message $m \in \mathcal{M}$ compute $s = d_{k_d}(m)$
receive (m, s)	$\longleftarrow$ signed message (m, s) $\longleftarrow$	transmit (m, s)
if $e_{k_e}(s) = m$ ACCEPT otherwise, REJECT		

One problem with this scheme is that it has effectively doubled the size of the message. The way to make a smaller, more efficient signature is for it to consist of the decryption not of all of m but instead of some function $h(m)$. Here the function h should take a message

of arbitrary size and produce a small, digested piece of data ... which nevertheless depends upon every part of the input m. After, all, if $h(m)$ depended only upon the first 100 bits of m, for example, then a malicious Eve could alter the message in transit, and her change would go undetected as long as she did not change the first 100 bits of the message.

Cryptologists have a name for functions like this h.

DEFINITION 4.5.1. A function h which takes as input arbitrary length strings of bits and produces output bit strings of a fixed length is called a **cryptographic hash function** if it satisfies

ease of computation: it is feasible to compute the $h(m)$ for any m;

pre-image resistance: given a hash value t, it is infeasible to find an m such that $h(m) = t$;

second pre-image resistance: given a specific input m_1, it is infeasible to find another m_2 such that $h(m_2) = h(m_1)$;

collision resistance: it is infeasible to find two messages m_1 and m_2 such that $h(m_2) = h(m_1)$.

The words *feasible* and *infeasible* here have the same meaning here as in the previous section that it is, or is not, possible to complete the computation in an amount of time bounded by a polynomial function of the size of the inputs.

Notice that since a hash function takes inputs of arbitrary length but has a fixed output size, there will necessarily be an infinite number of collisions

The creation of cryptographic hash functions is something of a black art. It turns out that if one builds a candidate hash function with some clear structure (usually mathematical) – particularly if it is one that is fast to compute – a way to break one of the resistance requirements is usually found by the cryptological community. For this reason, the algorithms currently in wide use tend to be very *ad hoc* computations that just seem messy and have resisted attempts at inversion or breaking resistance.

EXAMPLE 4.5.2. For around a decade starting in the early 1990s, the most widely used cryptographic hash function was called `md5`. This algorithm was developed by Ron Rivest and published in 1992. The output size of `md5` is 128 bits.

While `md5` was thought to be flawed since the middle 1990s, a real attack was not published until 2004, when it was shown not to be collision resistant [**WY05**]. However, `md5` is still used extensively today to verify that a large data transfer has not suffered a transmission error – *i.e.,* it is still a useful tool to test for non-malicious data corruption. (In this context of providing evidence for data integrity against non-malicious corruption, a hash function is frequently called a **fingerprint**.)

EXAMPLE 4.5.3. The most widely used cryptographic hash function from the late 1990s until recently, and one which is built into many widely accepted and standardized cryptographic protocols, is `SHA-1`, with an output size of 160 bits.

`SHA-1` was developed by US National Security Agency (NSA) in a semi-public process, and was adopted by the US National Institute of Standards and Technology (NIST) as part of several US Federal Information Processing Standards .

In 2004, some work was published which indicated that `SHA-1` might be vulnerable to certain kinds of attack. (See [**PS04**].) For this reason, NIST required in 2010 many US federal data protection applications to move to another hash function.

EXAMPLE 4.5.4. At the time of this writing, most security conscious users and organizations recommend `SHA-2`, usually in its "`SHA-256`" variant, which has an output size of 256 bits. Given recent revelations of the NSA's involvement – and weakening of – cryptographic protocols, it might be a cause of concern that NSA participated in the development of `SHA-2`.

Graphically:

RSA digital signatures:

Charlie	on public network	Bob
		do RSA set-up, make $k_e = (n, e)$ and d
download k_e	$\longleftarrow$ verification key k_e $\longleftarrow$	publish k_e
		message $m \in \mathcal{M}$
		compute $s = (h(m))^d$
receive (m, s)	$\longleftarrow$ signed message (m, s) $\longleftarrow$	transmit (m, s)
if $h(m) = s^e \pmod n$		
ACCEPT		
otherwise, REJECT		

With this understood, we can describe digital signatures more formally.

DEFINITION 4.5.5. Suppose Bob sets up the RSA cryptosystem, as in Definition 4.4.5, and chooses once and for all a cryptographic hash function h which he publishes on his web page along with his public encryption key k_e.

If Bob now wants to sign a message m, he appends to m the **RSA digital signature** $s = d_{d_k}(h(m))$ before transmitting it to any third party, say Charlie.

When Charlie receives a signed message (m, s) which claims to be from Bob, he goes to Bob's website and downloads the public key k_e and description of the cryptographic hash function h. At this point, Charlie computes $e_{k_e}(s)$ and compares it to $h(m)$. If they are equal, he **accepts** the signature; if not, he **rejects**.

In this context, Bob's private/decryption key k_d is called the **signing key** and his public/encryption key k_e is called the **verification key**.

Exercises for §4.5.

EXERCISE 4.5.1. Use the Pigeonhole Principle (Theorem 1.1.2) to prove that there will always be an infinite number of collisions for a cryptographic hash function [although they may not be feasibly computable].

EXERCISE 4.5.2. Give an example of a hash function h whose output size is one bit and a specific input m_1 for which there is no second pre-image; that is, $\nexists m_2$ such that $h(m_2) = h(m_1)$.

4.6. Man-in-the-Middle Attacks, Certificates, and Trust

While public-key crypto can seem an unalloyed benefit to the networked world, close examination of the details of the last two sections shows a dangerous gap between a casual statement of the properties of these cryptographic tools and their reality. The distinction which at first goes unnoticed is between *Bob, the person*, and *the bits* arriving over the wire to Alice or Larry *which claim to be from Bob*. This has little effect if Eve is, as we have mostly been assuming her to be, a passive observer of the communications between Alice and Bob (and sometimes Larry). But if Eve has even more control over the network and can replace all communications with her own versions, new attacks are possible.

Suppose Alice wants to send her secret message to Bob, again without ever having met him to exchange with him a key for a symmetric cryptosystem. She hopes that he has a public key, so she gets on the web and downloads his home page.

Here is where Eve springs into action, intercepting Bob's (web server's) response to the web request. She keeps a copy of Bob's public key k_e^B for herself, but substitutes into the web page data in its place an RSA encryption key of her own, k_e^E – for which she alone knows the corresponding decryption key k_d^E – and transmits the modified page on to Alice.

Alice composes her cleartext m, and transmits its corresponding ciphertext $c_A = e_{k_e^E}(m)$ to Bob, or so she thinks. Instead, Eve intercepts that ciphertext, decrypts and stores $m = d_{k_d^E}(c_A) = d_{k_d^E}(e_{k_e^E}(m))$. Then, in order to make Alice and Bob think everything is working normally (so they'll keep up their revealing chatter), Eve transmits $c_E = e_{k_e^B}(m)$ on to Bob.

From Bob's point of view, he gets an e-mail which seems to be from Alice and which, when decrypted with his private key, does make perfect sense. Furthermore, if he interacts with Alice off-line, she behaves as if she sent that message – in fact, she did, but not in the particular encrypted form that Bob received. Eve has completely violated the *confidentiality* of Alice and Bob's communications, and she could violate the message *integrity* any time she liked, still making it appear to come legitimately from Alice.

The above scenario is called a **man-in-the-middle attack** (pardon the non-gender-neutral terminology).

Here is a graphical depiction of this attack:

Man-in-the-middle attack:

Alice	Eve	Bob
	intercept $k_e^B \leftarrowtail$	generate keys: public k_e^B, private k_d^B publish k_e^B
download k_e^E	generate keys: public k_e^E, private k_d^E $\leftarrowtail k_e^E$, spoof origin	
message $m \in \mathcal{M}$ compute $c_A = e_{k_e^E}(m)$ transmit c_A	$\rightarrowtail c_A$, intercept	
	extract the cleartext $m = d_{k_d^E}(c_A)$ change to m' if desired compute $c_E = e_{k_e^B}(m')$ spoof origin $c_E \rightarrowtail$	receive c_E
		read the message $m' = d_{k_d^B}(c_E)$

So it seems that symmetric cryptosystems actually have one nice thing built in: when the parties meet in that perfect, prelapsarian moment to exchange their symmetric key, they presumably can have confidence in the identity of the person they're talking to – if not, they wouldn't exchange the symmetric key until they had seen lots of official-looking identification materials. Asymmetric cryptosystems have the fundamental difficulty to overcome of establishing a trustworthy connection between a real person's identity and a public key on the Internet which purports to be from that person.

The technique from last section, §4.5, can help transfer the trust, at least. Suppose Alice wants to engage in secret communication with Bob, but does not know if she can trust the public key which appears to be on Bob's web page. If that key came with an accompanying digital signature issued by a *trusted third party* [TTP] of which Alice had the public key, she could verify that the key was Bob's, at least as far as the TTP knew.

Here is a formal definition.

DEFINITION 4.6.1. Individuals or organizations who want to use asymmetric cryptography can go to a trusted third party called a **certificate authority** [**CA**] to request a [**digital**] **certificate** for their public keys. This certificate would be a digital signature on the public key itself, signed by the CA's signing key. The CA's verification key would be assumed widely disseminated across the Internet or perhaps built into the basic operating system software or computer hardware. Then anyone who wanted to use a public key could

first check the validity of the associated certificate and have confidence that the intended party did own the key in question.

The entire aggregate of certificates, CAs, pre-installed or widely distributed verification keys, *etc.*, is called a **public key infrastructure** or **PKI**.

In actual practice, the CA often cannot do more than certify that a certain public key is owned by an individual who has access to a certain e-mail address, or the webmaster password to a certain site, or some such purely Internet-based token. That much is usually quite easy – connecting this with a real, external identity would require checking some form of government-issued ID, probably, and is rarely done. Although it would be useful: perhaps the government should act as a CA, and government IDs should have a built-in RFID (recent US passports do!) which can transmit a certificate on the ID owner's public key.

There is one other approach to figuring out whether to have faith in a particular public key, favored by those who mistrust authority but are willing to trust some others they know personally. In this approach, individuals who know each other personally and have faith in each other's good cryptologic habits can each sign each other's public keys, adding their digital signatures to those which have already been collected. Then when you want to use someone's public key, you can unwrap a chain of digital signatures, each signing the next, until you find one by a person whom you know personally, have met, and whose public key you have verified.

This approach has become known as the **web of trust**, and is strongly supported by `GnuPG` and other `OpenPGP`-compatible organizations, see `gnupg.org`.

By the way, to be useful, a web of trust must include as many individuals and their keys as possible, and have as many connections (where individual X knows and signs individual Y's public key) as possible. One way to get this going quickly is to have a **key-signing party**. If you are standing next to someone whom you know well who says "sure, this other person is someone *I* know well and trust", then you might be willing to sign both of their keys right there. In practice, when you sign keys, you are standing there with someone whom you trust, so it usually suffices to check the `md5` fingerprint of their key and not the whole thing – the `md5` is shorter and standing there with this person you trust, presumably you do not think that anyone there has devoted large computational resources to finding a second `md5` pre-image the fingerprint.

CHAPTER 5

Indices = Discrete Logarithms

We talked about the *cyclic subgroup* $\langle a \rangle$ *of* $(\mathbb{Z}/n\mathbb{Z})^*$ *generated by an element* a in the proof of our version of Lagrange's Theorem 3.3.3, on the way to Euler's Theorem 3.3.4. Recall it consisted of all powers of a (well, of $[a]_n$) in $(\mathbb{Z}/n\mathbb{Z})^*$. Here are some examples:

n	$\phi(n)$	$(\mathbb{Z}/n\mathbb{Z})^*$	a	$\langle a \rangle$	$\mathrm{ord}_n(a)$
2	1	$\{1\}$	1	$\{1\}$	1
3	2	$\{1,\ 2\}$	1	$\{1\}$	1
			2	$\{2,\ 2^2 \equiv 1\}$	2
4	2	$\{1,\ 3\}$	1	$\{1\}$	1
			3	$\{3,\ 3^2 \equiv 1\}$	2
5	4	$\{1,\ 2,\ 3,\ 4\}$	1	$\{1\}$	1
			2	$\{2,\ 2^2 \equiv 4,\ 2^3 \equiv 3,\ 2^4 \equiv 1\}$	4
			3	$\{3,\ 3^2 \equiv 4,\ 3^3 \equiv 2,\ 3^4 \equiv 1\}$	4
			4	$\{4,\ 4^2 \equiv 1\}$	2
6	2	$\{1,\ 5\}$	1	$\{1\}$	1
			5	$\{5,\ 5^2 \equiv 1\}$	2
7	6	$\{1,\ 2,\ 3,\ 4,\ 5,\ 6\}$	1	$\{1\}$	1
			2	$\{2,\ 2^2 \equiv 4,\ 2^3 \equiv 1\}$	3
			3	$\{3,\ 3^2 \equiv 2,\ 3^3 \equiv 6,\ 3^4 \equiv 4,\ 3^5 \equiv 5,\ 3^6 \equiv 1\}$	6
			4	$\{4,\ 4^2 \equiv 2,\ 4^3 \equiv 1\}$	3
			5	$\{5,\ 5^2 \equiv 4,\ 5^3 \equiv 6,\ 5^4 \equiv 2,\ 5^5 \equiv 3,\ 5^6 \equiv 1\}$	6
			6	$\{6,\ 6^2 \equiv 1\}$	2
8	4	$\{1,\ 3,\ 5,\ 7\}$	1	$\{1\}$	1
			3	$\{3,\ 3^2 \equiv 1\}$	2
			5	$\{5,\ 5^2 \equiv 1\}$	2
			7	$\{7,\ 7^2 \equiv 1\}$	2

TABLE 5.0.1. Cyclic subgroups of $(\mathbb{Z}/n\mathbb{Z})^*$ for $n = 2, \ldots, 8$

Each of these cyclic subgroups $\langle a \rangle$ has a size – being the order $\mathrm{ord}_n(a)$ of its generator – which divides the size of the ambient $(\mathbb{Z}/n\mathbb{Z})^*$, as we know will happen from Euler's Theorem 3.3.4.

Some random things we also notice, which might or might not hold true in general, given the small amount of evidence in this table, are:

- frequently there is an element a which generates the biggest possible cyclic subgroup: $\langle a \rangle = (\mathbb{Z}/n\mathbb{Z})^*$;
- there seem always to be those big cyclic subgroups in $\mathbb{Z}/n\mathbb{Z}$ when n is a prime;
- even some composite n give a $\mathbb{Z}/n\mathbb{Z}$ which contains a big cyclic subgroup, except for the case of the largest power of 2, which was $n = 2^3$, that we tried.

Just to see if those possible general statements hold a bit further, let's compute two more examples. For these we give only the sizes of $(\mathbb{Z}/n\mathbb{Z})^*$ and $\langle a \rangle$, not complete lists of their elements, to conserve space and since the elements of $(\mathbb{Z}/n\mathbb{Z})^*$ can be found in the column labelled "a":

n	$\phi(n)$	$a \in (\mathbb{Z}/n\mathbb{Z})^*$	$\mathrm{ord}_n(a)$
16	8	1	1
		3	4
		5	4
		7	2
		9	2
		11	4
		13	4
		15	2
17	16	1	1
		2	8
		3	16
		4	4
		5	16
		6	16
		7	16
		8	8
		9	8
		10	16
		11	16
		12	16
		13	4
		14	16
		15	8
		16	2

TABLE 5.0.2. Cyclic subgroups of $(\mathbb{Z}/n\mathbb{Z})^*$ for $n = 16$ and $n = 17$

Our guesses (large powers of 2 not so good; other n's, particularly prime, are good) still hold true. Of course, any finite amount of evidence for a general mathematical statement is very inconclusive....

Now to the formal analysis.

5.1. More Properties of Multiplicative Order

But first, we need some more facts about [multiplicative] order.

THEOREM 5.1.1. *Suppose* $n \in \mathbb{N}$ *and* $a \in \mathbb{Z}$ *satisfy* $n \geq 2$ *and* $\gcd(a, n) = 1$. *Then* $a^k \equiv 1 \pmod{n}$ *for* $k \in \mathbb{N}$ *iff* $\mathrm{ord}_n(a) \mid k$.

PROOF. One direction is very easy: if $k \in \mathbb{N}$ satisfies $\mathrm{ord}_n(a) \mid k$ then $\exists d \in \mathbb{N}$ such that $k = \mathrm{ord}_n(a) \cdot d$ and thus

$$a^k = a^{\mathrm{ord}_n(a) \cdot d} = (a^{\mathrm{ord}_n(a)})^d \equiv 1^d = 1 \pmod{n}.$$

Conversely, suppose $k \in \mathbb{N}$ is such that $a^k \equiv 1 \pmod{n}$. Apply the Division Algorithm to get $q, r \in \mathbb{N}$ such that $k = \mathrm{ord}_n(a) \cdot q + r$ and $0 \leq r < \mathrm{ord}_n(a)$. Then

$$1 \equiv a^k = a^{\mathrm{ord}_n(a) \cdot q + r} = (a^{\mathrm{ord}_n(a)})^q \cdot a^r \equiv 1^q \cdot a^r \equiv a^r \pmod{n}.$$

But the definition of the order $\mathrm{ord}_n(a)$ is that it is the smallest positive integer such that a to that power is congruent to 1. The only way for $a^r \equiv 1 \pmod{n}$ and $0 \leq r < \mathrm{ord}_n(a)$ to be true, then, is if $r = 0$. Thus $k = \mathrm{ord}_n(a) \cdot q$ and $\mathrm{ord}_n(a) \mid k$, as desired. □

What this will mean is that when we work in the cyclic subgroup of $(\mathbb{Z}/n\mathbb{Z})^*$ generated by some element a, we should work with the powers of a as if the powers lived in $\mathbb{Z}/(\mathrm{ord}_n(a))\mathbb{Z}$. That is:

THEOREM 5.1.2. *Suppose* $n \in \mathbb{N}$ *and* $a \in \mathbb{Z}$ *satisfy* $n \geq 2$ *and* $\gcd(a, n) = 1$. *Then* $a^j \equiv a^k \pmod{n}$ *for* $j, k \in \mathbb{N}$ *iff* $j \equiv k \pmod{\mathrm{ord}_n(a)}$.

PROOF. Again, one direction is very easy: if $j, k \in \mathbb{N}$ satisfy $j \equiv k \pmod{\mathrm{ord}_n(a)}$ then $\exists \ell \in \mathbb{Z}$ such that $j = k + \mathrm{ord}_n(a) \cdot \ell$ from which we compute

$$a^j = a^{k + \mathrm{ord}_n(a) \cdot \ell} = a^k \cdot (a^{\mathrm{ord}_n(a)})^\ell \equiv a^k \cdot 1^\ell = a^k \pmod{n}.$$

Conversely, suppose $j, k \in \mathbb{N}$ are such that $a^j \equiv a^k \pmod{n}$ and, without loss of generality, $j \geq k$. Multiplying both sides of this congruence by $(a^{-1})^k$, which exists since $\gcd(a, n) = 1$, we get $a^{j-k} \equiv 1 \pmod{n}$. Then by Theorem 5.1.1 we have $\mathrm{ord}_n(a) \mid (j - k)$ or $j \equiv k \pmod{\mathrm{ord}_n(a)}$. □

EXAMPLE 5.1.3. The rows in Table 5.0.1 bear out Theorems 5.1.1 and 5.1.2: each cyclic subgroup (row) has a number of elements which divides the corresponding $\phi(n)$, and powers of the generator a are only defined up to $\mathrm{ord}_n(a)$.

It also seems that some of the smaller cyclic subgroups sometimes occur as a subset of a larger cyclic subgroup. So in mod $n = 7$, if $a = 3$ and $b = a^2 \equiv 2$, then we have the containment $\langle b \rangle \subset \langle a \rangle$, where $\langle b \rangle$ consists of half of the elements of $\langle a \rangle$, namely the even powers of a:

$$\langle b \rangle = \left\{b,\ b^2,\ b^3\right\} = \{2,\ 4,\ 1\} = \left\{3^2,\ 3^4,\ 3^6\right\} \subset \left\{3,\ 3^2,\ 3^3,\ 3^4,\ 3^5,\ 3^6\right\} = \langle a \rangle$$

Looking instead at $c = 3^3 \equiv 6$, we still have $\langle c \rangle \subset \langle a \rangle$, but now $\langle c \rangle$ consists of one third of the elements of $\langle a \rangle$, the powers of a which are multiples of 3:

$$\langle c \rangle = \left\{c,\ c^2\right\} = \{6,\ 1\} = \left\{3^3,\ 3^6\right\} \subset \left\{3,\ 3^2,\ 3^3,\ 3^4,\ 3^5,\ 3^6\right\} = \langle a \rangle$$

The last part of this example seems to be asking for a general statement about what size subset of a cyclic subgroup $\langle a \rangle$ is formed of the cyclic subgroup $\left\langle a^k \right\rangle$ for some $k \in \mathbb{N}$. This size will just be the order of that element a^k, so we need the following

THEOREM 5.1.4. *Suppose* $n \in \mathbb{N}$ *and* $a \in \mathbb{Z}$ *satisfy* $n \geq 2$ *and* $\gcd(a, n) = 1$. *Then* $\forall k \in \mathbb{N}$, $ord_n(a^k) = \frac{\mathrm{ord}_n(a)}{\gcd(\mathrm{ord}_n(a), k)}$

PROOF. Fix some $k \in \mathbb{N}$ and let $r = \mathrm{ord}(a^k)$ and $s = \mathrm{ord}(a)$; with this notation, what we are trying to prove is that $r = \dfrac{s}{\gcd(s, k)}$.

We shall repeatedly use in the proof the fact that since a gcd is a divisor, $\gcd(s, k) \mid s$ and $\gcd(s, k) \mid k$, which means in turn that both $\dfrac{s}{\gcd(s, k)}, \dfrac{k}{\gcd(s, k)} \in \mathbb{N}$.

Now to the proof.

The definition of order is that r is the smallest natural number such that $(a^k)^r \equiv 1 \pmod{n}$. Notice that

$$(a^k)^{\frac{s}{\gcd(s,k)}} = (a^s)^{\frac{k}{\gcd(s,k)}} = 1^{\frac{k}{\gcd(s,k)}} \equiv 1 \pmod{n}.$$

By Theorem 5.1.1, we conclude that $r \mid \left(\dfrac{s}{\gcd(s, k)}\right)$.

Smallest or not, since $a^{kr} = (a^k)^r \equiv 1 \pmod{n}$, $s \mid kr$ by Theorem 5.1.1, so $\exists m \in \mathbb{N}$ such that $ms = kr$. Dividing both sides by $\gcd(s, k)$, we get an equation of natural numbers

$$m\left(\frac{s}{\gcd(s, k)}\right) = \left(\frac{k}{\gcd(s, k)}\right) r;$$

which means

$$\frac{s}{\gcd(s, k)} \mid \left(\frac{k}{\gcd(s, k)}\right) r\ .$$

By Theorem 1.5.5, $\gcd\left(\dfrac{s}{\gcd(s, k)}, \dfrac{k}{\gcd(s, k)}\right) = 1$, so Euclid's Lemma 2.1.6 tells us that $\dfrac{s}{\gcd(s, k)} \mid r$.

We have shown that r and $\dfrac{s}{\gcd(s, k)}$ divide each other. But that means they are equal, which is what we were trying to prove. □

Exercises for §5.1.

EXERCISE 5.1.1. Suppose $n \in \mathbb{N}$ satisfies $n \geq 2$. Prove that if there exists $a \in (\mathbb{Z}/n\mathbb{Z})^*$ such that $\mathrm{ord}_n(a) = n - 1$, then n is prime.

EXERCISE 5.1.2. Suppose p is an odd prime and $a \in (\mathbb{Z}/p\mathbb{Z})^*$. Prove that if $\exists k \in \mathbb{N}$ such that $\mathrm{ord}_p(a) = 2k$ then $a^k \equiv -1 \pmod{p}$. *[Hint: look at the proof of Lemma 3.2.1.]*

EXERCISE 5.1.3. Suppose $n \in \mathbb{N}$ satisfies $n \geq 2$ and $a, b \in \mathbb{Z}/n\mathbb{Z}$ are both relatively prime to n. Prove that

$$\mathrm{ord}_n(ab) \mid \mathrm{ord}_n(a)\,\mathrm{ord}_n(b)\ .$$

EXERCISE 5.1.4. Prove that there are infinitely many primes congruent to 1 mod 4, by filling in the details of the following outline:

(1) Prove: if an odd prime p and $n \in \mathbb{Z}$ are such that $n^2 \equiv -1 \pmod{p}$, then $4 \mid \phi(p)$ *[use a theorem in this section]*. Why is it necessary to exclude the even prime here?
(2) Therefore for $n \in \mathbb{Z}$, the odd prime divisors of $n^2 + 1$ are congruent to 1 mod 4.
(3) Now assume for contradiction's sake that there are only finitely many primes $p_1, \ldots, p_n$ congruent to 1 mod 4 and consider the number $(2p_1 \cdot \cdots \cdot p_n)^2 + 1$. Apply the previous step....

5.2. A Necessary Digression: Gauss's Theorem on Sums of Euler's Function

Later in this chapter we will need a fact first proved by Gauss about Euler's ϕ function:

THEOREM 5.2.1. *For all* $n \in \mathbb{N}$, $\displaystyle\sum_{\substack{d\in\mathbb{N}\\ s.t.\ d|n}} \phi(d) = n$.

We'll give two proofs, which illustrate different features of this situation:

PROOF 1. Fix $n \in \mathbb{N}$.

Let's define some subsets of $\{1, \ldots, n\}$, dependent upon a choice of a positive divisor $d \mid n$, as follows

$$S_d = \{k \in \mathbb{N} \mid 1 \leq k \leq n \text{ and } \gcd(k, n) = d\} \ .$$

These sets are disjoint since for each $k \in \mathbb{N}$ such that $1 \leq k \leq n$, $d = \gcd(k, n)$ has a specific value and k is only in that S_d.

For $k \in S_d$, $\gcd(k, n) = d$ or, by Theorem 1.5.5, $\gcd(k/d, n/d) = 1$. Thus $\#(S_d)$ is the number of elements $\ell \in \mathbb{N}$ in the range $1 \leq \ell \leq n/d$ which are relatively prime to n/d, *i.e.,* $\#(S_d) = \phi(n/d)$.

Every $k \in \mathbb{Z}$ in the range $1 \leq k \leq n$ is in exactly one of these S_d, for d a positive divisor of n. Therefore

$$\{1, \ldots, n\} = \bigcup_{\substack{d\in\mathbb{N}\\ s.t.\ d|n}} S_d$$

so

$$n = \#\left(\{1, \ldots, n\}\right) = \#\left(\bigcup_{\substack{d\in\mathbb{N}\\ s.t.\ d|n}} S_d\right) = \sum_{\substack{d\in\mathbb{N}\\ s.t.\ d|n}} \#\left(S_d\right) = \sum_{\substack{d\in\mathbb{N}\\ s.t.\ d|n}} \phi(n/d) \ .$$

But as d runs over the positive divisors of n, so does n/d; in other words

$$\{d \mid d \in \mathbb{N},\ 1 \leq d \leq n \text{ and } d \mid n\} = \{n/d \mid d \in \mathbb{N},\ 1 \leq d \leq n \text{ and } d \mid n\}$$

We can therefore rewrite the last big equation as

$$n = \sum_{\substack{d\in\mathbb{N}\\ s.t.\ d|n}} \phi(n/d) = \sum_{\substack{d\in\mathbb{N}\\ s.t.\ d|n}} \phi(d)$$

which is what we were trying to prove. □

PROOF 2. This approach will concentrate on the function

$$F(n) = \sum_{\substack{d\in\mathbb{N}\\ s.t.\ d|n}} \phi(d) \ ,$$

which has some very nice properties.

For example, suppose p is a prime and $k \in \mathbb{N}$. Then the divisors of p^k are $1, p, p^2, \ldots, p^k$, so

$$F(p^k) = \phi(1)+\phi(p)+\phi(p^2)+\cdots+\phi(p^k) = 1+(p-1)+(p^2-p)+\cdots+(p^k-p^{k-1}) = p^k \ .$$

Furthermore, if p and q are distinct primes, then the divisors of pq are 1, p, q, and pq. Also, $\gcd(p,q) = 1$, so by Theorem 2.5.3

$$\begin{aligned} F(pq) &= \phi(1) + \phi(p) + \phi(q) + \phi(pq) \\ &= \phi(1) + \phi(p) + \phi(q) + \phi(p)\phi(q) \\ &= (1 + \phi(p))(1 + \phi(q)) \\ &= F(p)F(q) \ , \end{aligned}$$

From this fact, one can prove (it's an exercise below) that $F(ab) = F(a)F(b)$ whenever $a, b \in \mathbb{N}$ are relatively prime. This means that F has the same sort of multiplicative property for relatively prime factors as does ϕ.

We are ready to finish the proof. So let $n \in \mathbb{N}$ be any number such that $n > 1$ (for $n = 1$, the theorem is trivial). Suppose the prime decomposition of n is given by

$$n = p_1^{k_1} \cdot \cdots \cdot p_r^{k_r} \ ,$$

where the primes $\{p_1, \ldots, p_r\}$ are distinct. Therefore $\gcd(p_i^{k_i}, p_j^{k_j}) = 1$ if $i \neq j$ and so we can use the multiplicative property of F and our calculation of F for prime powers to conclude

$$\begin{aligned} F(n) &= F(p_1^{k_1} \cdot \cdots \cdot p_r^{k_r}) \\ &= F(p_1^{k_1}) \cdot \cdots \cdot F(p_r^{k_r}) \\ &= p_1^{k_1} \cdot \cdots \cdot p_r^{k_r} \\ &= n \ , \end{aligned}$$

which is what we wanted to prove. □

Exercises for §5.2.

EXERCISE 5.2.1. Finish the second proof of Gauss's Theorem 5.2.1 by proving that $F(ab) = F(a)F(b)$ for all $a, b \in \mathbb{N}$ which are relatively prime.

5.3. Primitive Roots

Now back to those elements which generate large cyclic subgroups – which have a name:

DEFINITION 5.3.1. Given $n \in \mathbb{N}$ such that $n \geq 2$, an element $a \in (\mathbb{Z}/n\mathbb{Z})^*$ is called a **primitive root mod** n if $\mathrm{ord}_n(a) = \phi(n)$. We shall also call an integer $x \in \mathbb{Z}$ a **primitive root mod** n if $[x]_n$ is a primitive root in the sense just defined.

EXAMPLE 5.3.2. From the two tables in the introduction to this chapter we can read off the following primitive roots mod their respective n's:

n	Primitive Roots mod n	$\phi(n)$	$\phi(\phi(n))$
2	1	1	1
3	2	2	1
4	3	2	1
5	2, 3	4	2
6	5	2	1
7	3, 5	6	2
8	*none*	4	2
⋮			
16	*none*	8	4
17	3, 5, 6, 7, 10, 11, 12, 14	16	8

TABLE 5.3.1. Primitive roots for $n = 2, \ldots, 8, 16, 17$

We have included the column of $\phi(n)$ since that is the order that each primitive root must have. And then we added the column of $\phi(\phi(n))$ as well, since by some strange magic it appears frequently to compute the number of primitive roots.

Let us state formally, and prove, the general result noticed in this example:

THEOREM 5.3.3. *Given $n \in \mathbb{N}$ satisfying $n \geq 2$, if n has any primitive roots than it has exactly $\phi(\phi(n))$ primitive roots.*

PROOF. Let $a \in (\mathbb{Z}/n\mathbb{Z})^*$ be a primitive root. That means that

$$\langle a \rangle = \left\{a,\ a^2,\ \ldots\ a^{\mathrm{ord}_n(a)}\right\} = (\mathbb{Z}/n\mathbb{Z})^*$$

since $\mathrm{ord}_n(a) = \phi(n) = \#\left((\mathbb{Z}/n\mathbb{Z})^*\right)$. Any other primitive root b, being an element of $(\mathbb{Z}/n\mathbb{Z})^*$ must be one of these powers of a. In other words, $\exists k \in \mathbb{N}$ such that $b = a^k$.

In order for this b to be a primitive root, its order must be $\phi(n)$. But from Theorem 5.1.4, we know that

$$\mathrm{ord}(b) = \frac{\mathrm{ord}(a)}{\gcd(\mathrm{ord}(a), k)} = \frac{\phi(n)}{\gcd(\phi(n), k)}\ .$$

Therefore $b = a^k$ will be a primitive root if and only if $\gcd(\phi(n), k) = 1$. From the definition of Euler's ϕ function, this happens for exactly $\phi(\phi(n))$ values of k. □

Let's see if we can prove that in some circumstances, we will have the first primitive root that the above theorem requires. The easiest situation in which that might happen will probably be when the modulus is prime, as we have the strongest tools to work with in that case.

Finding elements of a particular order d amounts (in part) to finding solutions to the equation $x^d - 1 \equiv 0 \pmod{p}$. The first step toward this is the following more general theorem about polynomials in mod p due to Lagrange:

THEOREM 5.3.4. *For $n \in \mathbb{N}$ and $a_0, \ldots, a_n \in \mathbb{Z}$, the polynomial*

$$a_n x^n + \cdots + a_1 x + a_0 \equiv 0 \pmod{p} ,$$

where $a_n \not\equiv 0 \pmod{p}$, has at most n solutions in $\mathbb{Z}/p\mathbb{Z}$ if p is prime.

PROOF. We use induction on the degree n. The base case $n = 1$ amounts to solving the linear congruence $a_1 x + a_0 \equiv 0 \pmod{p}$ where $a_1 \not\equiv 0 \pmod{p}$. Since p is prime, this means that $\gcd(p, a_1) = 1$, and therefore the linear congruence has a unique solution mod p by the version of Theorem 2.2.4 as stated in Remark 2.2.5.

Now for the inductive step, assuming that the theorem is true for some $n \in \mathbb{N}$, we shall prove it for the case $n + 1$. So let $a_0, \ldots, a_{n+1} \in \mathbb{Z}$ be such that $a_{n+1} \not\equiv 0 \pmod{p}$. If

$$a(x) = a_{n+1} x^{n+1} + a_n x^n + \cdots + a_1 x + a_0$$

has no zeros in mod p, then the theorem is certainly true for this polynomial of degree $n+1$: the number of solutions to $a(x) \equiv 0 \pmod{p}$ is $0 < n + 1$.

If instead $a(x)$ has at least one zero z_1 in mod p, do long division of polynomials to get

$$a(x) = b(x)(x - z_1) + r(x)$$

where $b(x)$ and $r(x)$ are polynomials with integral coefficients and such that

$$0 \leq \deg(r(x)) < \deg(x - z_1) = 1 .$$

It therefore follows that $r(x)$ is a constant polynomial, say with value $r_1 \in \mathbb{Z}$.

Plug z_1 into the above formula resulting from division of polynomials to get

$$0 \equiv a(z_1) \equiv b(z_1)(z_1 - z_1) + r_1 \equiv r_1 \pmod{p} .$$

Hence $r_1 \equiv 0 \pmod{p}$, which means that our polynomial $a(x) \equiv b(x)(x - z_1) \pmod{p}$. But a_{n+1} equals (one times) the leading coefficient of $b(x)$, so that leading coefficient must not be congruent to 0 mod p.

Hence the inductive hypothesis applies to $b(x)$ and tells us that it has no more than n roots in mod p. These roots of $b(x)$, plus the single root of $(x - z_1)$, total no more than $n + 1$ roots for $a(z) = b(x)(x - z_1)$.

All we need to do to finish is to see that any root of $a(x)$ must in fact be a root of $b(a)$ or $(x - z_1)$. So suppose $r \in \mathbb{Z}$ is a root, so

$$a(r) = b(r)(r - z_1) \equiv 0 \pmod{p},$$

which means in turn that $p \mid b(r)(r - z_1)$. Then by Proposition 3.1.5 we must have $p \mid b(r)$ or $p \mid (r - z_1)$. In other words, $b(r) \equiv 0 \pmod{p}$ or $(r - z_1) \equiv 0 \pmod{p}$, as we hoped. [This amounts to using what is called in basic algebra the "zero product property", which is true in $\mathbb{Z}/p\mathbb{Z}$ for p prime by Proposition 3.1.5; in abstract algebra, we say that $\mathbb{Z}/p\mathbb{Z}$ is a *domain* if p is prime.]

Thus the roots of $a(x)$ all come from roots of $b(x)$ or $(x - z_1)$, and so number at most $n + 1$, which means we have proven the inductive step. □

So much for a maximum number of roots. In the following particular case, we can get the exact number of roots:

THEOREM 5.3.5. *If p is prime and $d \mid p - 1$, then there are exactly d solutions, up to congruence mod p, of the congruence*

$$x^d = 1 \pmod{p}.$$

PROOF. Let p and d be as in the statement, and define $m = (p - 1)/d \in \mathbb{N}$. We use a clever factorization, defining

$$\begin{aligned} a(x) &= x^d - 1, \\ b(x) &= x^{dm-d} + x^{dm-2d} + \cdots + x^d + 1, \text{ and} \\ c(x) &= x^{p-1} - 1 \end{aligned}$$

so that $c(x) = a(x)b(x)$.

Notice that by Fermat's Little Theorem, $c(x)$ has exactly $p - 1$ roots which are distinct in mod p, being $\{1, \ldots, p - 1\}$. Also, by the previous Theorem 5.3.4, $b(x)$ has at most $\deg(b(x)) = dm - d = p - 1 - d$ roots. Since, as in the proof of that Theorem 5.3.4 (the part where we mentioned the "zero product property"), roots of $c(x)$ correspond exactly to the roots of $a(x)$ and those of $b(x)$, there must be at least d roots of $a(x)$ to add to these no more than $p - 1 - d$ roots of $b(x)$, making the exactly $p - 1$ roots of $p - 1$. □

We are now in a position to quantify exactly the congruence classes in $\mathbb{Z}/p\mathbb{Z}$, for p a prime, of particular orders:

THEOREM 5.3.6. *If p is prime and $d \mid p - 1$, then there are exactly $\phi(d)$ distinct congruence classes of order d in $\mathbb{Z}/p\mathbb{Z}$.*

PROOF. Let p and d be as in the statement and write

$$\psi(k) = \#\left(\{\ell \in \mathbb{Z} \mid 1 \leq \ell \leq p - 1 \text{ and } \operatorname{ord}_p(\ell) = k\}\right).$$

By our version of Lagrange's Theorem 3.3.3, any $\ell \in \mathbb{Z}$ such that $1 \leq \ell \leq p-1$ has an order which divides $\phi(p) = p-1$, so

$$\sum_{\substack{k \in \mathbb{N} \\ s.t.\ k|p-1}} \psi(k) = p-1 \ .$$

In addition, by Gauss's Theorem 5.2.1,

$$\sum_{\substack{k \in \mathbb{N} \\ s.t.\ k|p-1}} \phi(k) = p-1 \ .$$

Therefore

$$\sum_{\substack{k \in \mathbb{N} \\ s.t.\ k|p-1}} \phi(k) = \sum_{\substack{k \in \mathbb{N} \\ s.t.\ k|p-1}} \psi(k) \ .$$

Our goal now is to show that $\forall k \in \mathbb{N}$ such that $k \mid p-1$, we have $\psi(k) \leq \psi(k)$. If we can do this, then in fact for all such k, we would have to have $\psi(k) = \psi(k)$ because if for one k we had $\psi(k) < \psi(k)$ then there would be no way for another k to give $\psi(k) > \psi(k)$ so that $\sum \psi(k) = \sum \phi(k)$ could still hold.

So let $k \in \mathbb{N}$ satisfy $k \mid p-1$. If $\psi(k) = 0$, meaning that there are no elements of order k, then certainly $\psi(k) \leq \phi(k)$ as $\phi(k)$ is a non-negative function.

Suppose $\psi(k) > 0$, meaning there exists at least one element, call it a, of order k in $\mathbb{Z}/p\mathbb{Z}$. Notice that for any $n \in \mathbb{N}$, $(a^n)^k = (a^k)^n \equiv 1^n \equiv 1 \pmod{p}$, which means that $a, \ldots, a^k$ are distinct elements of $\mathbb{Z}/p\mathbb{Z}$ which all satisfy

$$x^k - 1 \equiv 1 \pmod{p} \ .$$

By Theorem 5.3.5, there are exactly k solutions of this congruence equation. These solutions include all of the elements of $\mathbb{Z}/p\mathbb{Z}$ of order k, and maybe other elements, whose order is a divisor of k, in which we are not so interested.

But in fact, by Theorem 5.1.4, we know exactly the orders of these elements $a, \ldots, a^k$: the order of a^ℓ is

$$\frac{\mathrm{ord}_p(a)}{\gcd(\mathrm{ord}_p(a), \ell)} = \frac{k}{\gcd(k, \ell)} \ .$$

Thus the elements of $\mathbb{Z}/p\mathbb{Z}$ of order k are those elements a^ℓ for $\ell \in \mathbb{Z}$ satisfying $1 \leq \ell \leq k$ for which $\gcd(k, \ell) = 1$. There are therefore $\phi(k)$; in other words, $\psi(k) = \phi(k)$. □

So, the punch line:

COROLLARY 5.3.7. *If p is prime, there are $\phi(p-1)$ primitive roots in $\mathbb{Z}/p\mathbb{Z}$.*

PROOF. Use $k = p-1$ in the previous theorem. □

Just to finish the thread of conjectures with which we started this chapter and section, let us prove the

THEOREM 5.3.8. *Let $k \in \mathbb{N}$ satisfy $n \geq 3$. Then $n = 2^k$ has no primitive roots.*

PROOF. We shall prove that for all odd numbers a,

$$a^{2^{k-2}} \equiv 1 \pmod{2^k}$$

by induction on k.

Start with $k = 3$ and just look at all the congruence classes in $\mathbb{Z}/8\mathbb{Z}$ with odd representatives, and the squares of these classes:

$$[1]_8^2 = [1]_8, \qquad [3]_8^2 = [9]_8 = [1]_8, \qquad [5]_8^2 = [25]_8 = [1]_8, \quad \text{and} \quad [7]_8^2 = [49]_8 = [1]_8 .$$

So the base case $k = 3$ is established.

Now assume the statement holds for the value k, meaning that for all odd a,

$$a^{2^{k-2}} \equiv 1 \pmod{2^k} .$$

In other words, $\exists \ell \in \mathbb{Z}$ such that

$$a^{2^{k-2}} = 2^k \ell + 1 .$$

Squaring both sides of this, we get

$$a^{2^{k-1}} = \left(a^{2^{k-2}}\right)^2 = \left(2^k \ell + 1\right)^2 = 2^{2k}\ell^2 + 2 \cdot 2^k \ell + 1 = 2^{k+1}\left(2^{k-1}\ell^2 + \ell\right) + 1,$$

meaning that

$$a^{2^{(k+1)-2}} = a^{2^{k-1}} \equiv 1 \pmod{2^{(k+1)}} .$$

The inductive proof of the statement with which we began this proof is done. But notice that this means that for all odd numbers a, or, otherwise, for all $a \in (\mathbb{Z}/2^k\mathbb{Z})^*$, $\mathrm{ord}_{2^k}(a) \leq 2^{k-2} < 2^{k-1}$. Thus $n = 2^k$ has no primitive roots, which would all have to be odd numbers of order $\phi(2^k) = 2^{k-1}$. □

As we thought we noticed, based on the few examples we had calculated, large powers of two do not have primitive roots.

Exercises for §5.3.

EXERCISE 5.3.1. Express each of the primitive roots of 17 as a power of one of them.

EXERCISE 5.3.2. Find all of the primitive roots for the primes 11 and 13 and express them each as a power of one of them.

EXERCISE 5.3.3. Find all of the elements of $\mathbb{Z}/13\mathbb{Z}$ which have each possible order.

EXERCISE 5.3.4. By expressing everything as powers of single primitive root, use Corollary 5.3.7 to prove one direction of Wilson's Theorem

EXERCISE 5.3.5. If r is a primitive root of the odd prime p, prove that $r^{(p-1)/2} \equiv -1 \pmod{p}$. Also, prove that if s is any other primitive root of p, then rs cannot be a primitive root.

EXERCISE 5.3.6. Prove that the inverse of a primitive root is always a primitive root.

EXERCISE 5.3.7. If p is a prime congruent to $1 \mod 4$, prove that for all primitive roots r of p, $-r$ is also a primitive root. If instead p is a prime congruent to $3 \mod 4$, prove that $\mathrm{ord}_p(-r) = (p-1)/2$.

5.4. Indices

So for some values of $n \in \mathbb{N}$, there exists a primitive root $a \in (\mathbb{Z}/n\mathbb{Z})^*$, in which case we have seen that

$$\langle a \rangle = \left\{ a,\ a^2,\ \ldots\ a^{\mathrm{ord}_n(a)} \right\} = (\mathbb{Z}/n\mathbb{Z})^* \ .$$

That means that any $b \in (\mathbb{Z}/n\mathbb{Z})^*$ is some power of a. "But which power?" you cry, so we make the

DEFINITION 5.4.1. If $n \in \mathbb{N}$ has a primitive root a, then for any $b \in \mathbb{Z}$ such that $\gcd(b, n) = 1$ the smallest $k \in \mathbb{N}$ such that $b \equiv a^k \pmod{n}$ is called the **index of** b **relative to** a and is denoted $\mathrm{ind}_a(b)$. We shall also sometimes talk about $\mathrm{ind}_a(x)$ where $x \in (\mathbb{Z}/n\mathbb{Z})^*$, meaning the index relative to a of any representative b of the congruence class x.

EXAMPLE 5.4.2. Working from the tables above (5.0.1, 5.0.2, and 5.3.1), it is easy to compute a number of examples. First, for the smaller values of the moduli, where there are few primitive roots:

n	a	b	$\mathrm{ind}_a(b)$
2	1	1	1
3	2	1	2
		2	1
4	3	1	2
		3	1
5	2	1	4
		2	1
		3	3
		4	2
5	3	1	4
		2	3
		3	1
		4	2

n	a	b	$\mathrm{ind}_a(b)$
7	3	1	6
		2	2
		3	1
		4	4
		5	5
		6	3
	5	1	6
		2	4
		3	5
		4	2
		5	1
		6	3

TABLE 5.4.1. Index values for moduli $n = 2,\ 3,\ 4,\ 5,\ 7$

For the two larger moduli we worked out above, only $n = 17$ has primitive roots. Since $(\mathbb{Z}/17\mathbb{Z})^*$ has 16 elements and 8 primitive roots, we make a larger table for just this modulus:

$\mathrm{ind}_a(b)$		b															
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
a	3	16	14	1	12	5	15	11	10	2	3	7	13	4	9	6	8
	5	16	6	13	12	1	3	15	2	10	7	11	9	4	5	14	8
	6	16	2	15	4	11	1	5	6	14	13	9	3	12	7	10	8
	7	16	10	3	4	15	13	1	14	6	9	5	7	12	11	2	8
	10	16	10	11	4	7	5	9	14	6	1	13	15	12	3	2	8
	11	16	2	7	4	3	9	13	6	14	5	1	11	12	15	10	8
	12	16	6	5	12	9	11	7	2	10	15	3	1	4	13	14	8
	14	16	14	9	12	13	7	3	10	2	11	16	5	4	1	6	8

TABLE 5.4.2. Index values for modulus $n = 17$

Some things are natural to conjecture, given these examples and the simple definition of index; many of them are very easy to prove (and are exercises):

THEOREM 5.4.3. *Let $n \in \mathbb{N}$ have a primitive root a. Then*

(1) $\mathrm{ind}_a(1) = \mathrm{ord}_n(a) = \phi(n)$*; equivalently,* $\mathrm{ind}_a(1) \equiv 0 \pmod{\phi(n)}$
(2) $\mathrm{ind}_a(a) = 1$
(3) $\forall b, c \in \mathbb{Z}$ *such that* $\gcd(b, n) = \gcd(c, n) = 1$*, we have*

$$\mathrm{ind}_a(bc) \equiv \mathrm{ind}_a(b) + \mathrm{ind}_a(c) \pmod{\phi(n)}$$

(4) $\forall b \in \mathbb{Z}$ *such that* $\gcd(b, n) = 1$ *and* $\forall k \in \mathbb{N}$*, we have*

$$\mathrm{ind}_a(b^k) \equiv k \cdot \mathrm{ind}_a(b) \pmod{\phi(n)}$$

These properties of ind_a are strikingly similar to the basic properties of a logarithm with base a. For this reason, indices are called **discrete logarithms** in the computer science literature. For cryptological applications, it is important to note that exponentiation in some $(\mathbb{Z}/n\mathbb{Z})^*$ is an excellent candidate one-way function:

- Given n, a, and k, fast modular exponentiation is a feasible computation of $b = a^k$ in mod n.
- Given n, a, and b, no known feasible algorithm finds a $k = \mathrm{ind}_a(b)$ such that $a^k \equiv b \pmod{n}$.

We will discuss some discrete logarithm-based cryptological protocols in the next sections.

Before turning to cryptology, we explore some pure mathematical applications of indices. Like the applications of logarithms in basic algebra, the usefulness of indices comes from the above Theorem 5.4.3 enabling convenient algebraic manipulations of powers and multiplications. Here's an

EXAMPLE 5.4.4. We use indices to solve the congruence

$$3x^5 \equiv 4 \pmod{7}$$

by first taking ind_5 of both sides

$$\text{ind}_5(3) + 5\,\text{ind}_5(x) \equiv \text{ind}_5(4) \pmod 6$$

and applying all the rules in Theorem 5.4.3. Looking in Table 5.4.1, we translate that into

$$5 + 5 \cdot \text{ind}_5(x) \equiv 2 \pmod 6$$

or, solving:

$$\text{ind}_5(x) \equiv 5^{-1} \cdot 3 \equiv 5 \cdot 3 \equiv 3 \pmod 6$$

which means, searching through the table again, that $x = 6$.

Just to check, notice that $3 \cdot 6^5 = 23328 \equiv 4 \pmod 7$.

What if we solve the same equation, but use a different primitive root in our indices? Compute

$$\text{ind}_3(3) + 5\,\text{ind}_3(x) \equiv \text{ind}_3(4) \pmod 6$$

so

$$1 + 5 \cdot \text{ind}_5(x) \equiv 4 \pmod 6$$

and this yields the same equation

$$\text{ind}_5(x) \equiv 5^{-1} \cdot 3 \equiv 3 \pmod 6$$

and thus the same solution $x = 6$.

Caution: A common mistake with indices is to use the same modulus with the indices as with the original congruence. Instead, when the congruence is in mod n, for $n \in \mathbb{N}$, the index congruence is in mod $\phi(n)$!

Exercises for §5.4.

EXERCISE 5.4.1. In the modulus $n = 17$, use indices to solve the congruences

(a) $4x \equiv 11$ **(b)** $5x^6 \equiv 7$ **(c)** $x^{12} \equiv 13$
(d) $8x^5 \equiv 10$ **(e)** $9x^8 \equiv 8$ **(f)** $7^x \equiv 7$

EXERCISE 5.4.2. The logarithm rules in Theorem 5.4.3 are very similar to rules for the usual logarithm, except one is missing: the change of base formula. Figure out what that should be in the context of indices, make a formal statement, and prove it.

EXERCISE 5.4.3. Let p be an odd prime and a a primitive root mod p.

(a) Prove that $\operatorname{ind}_a(-1) = (p-1)/2$
(b) If $x, y \in (\mathbb{Z}/p\mathbb{Z})^*$ satisfy $xy \equiv 1 \pmod{p}$, then what is the relationship between $\operatorname{ind}_a(x)$ and $\operatorname{ind}_a(y)$? Prove it!
(c) If $x, y \in (\mathbb{Z}/p\mathbb{Z})^*$ satisfy $x + y \equiv 0 \pmod{p}$, then what is the relationship between $\operatorname{ind}_a(x)$ and $\operatorname{ind}_a(y)$? Prove it!

5.5. Diffie-Hellman Key Exchange

About a year before the RSA cryptosystem was invented, Whitfield Diffie and Martin Hellman published *New directions in cryptography* [**DH76**], the first full description of a working public key cryptosystem in the open scientific literature.[1] They defined in this paper something which has since been named after them:

DEFINITION 5.5.1. The following protocol is called **Diffie-Hellman key exchange** [**DHKE**]:

(1) Alice and Bob agree upon a large prime p and a primitive root $r \in (\mathbb{Z}/p\mathbb{Z})^*$ and publish both.
(2) Alice chooses an $\alpha \in \mathbb{Z}$ satisfying $1 \leq \alpha \leq p-1$, computes $A = r^\alpha \pmod p$, keeps α secret, but publishes A.
(3) Bob chooses a $\beta \in \mathbb{Z}$ satisfying $1 \leq \beta \leq p-1$, computes $B = r^\beta \pmod p$, keeps β secret, and publishes B.
(4) Alice gets the public value B and computes $S = B^\alpha$.
(5) Bob gets A and computes the same value $S = A^\beta$.
(6) Both Alice and Bob use the shared secret S for future communications encrypted with some symmetric cryptosystem upon which they had previously agreed.

The value S which both Alice and Bob have [but Eve does not] is called their **shared key** or **shared secret**.

Here is a graphical representation:

Diffie-Hellman key exchange:

Alice	on public network	Bob
	pick a prime p find a primitive root r	
pick α, compute $A = r^\alpha \pmod p$ publish A	$\rightarrowtail A \qquad B \leftarrowtail$	pick β, compute $B = r^\beta \pmod p$ publish B
get B, compute $S = B^\alpha \pmod p$		get A, compute $S = A^\beta \pmod p$
$\longleftarrow$ assymteric crypto with key S $\longrightarrow$		

PROPOSITION 5.5.2. *When Alice and Bob follow the DHKE protocol, they both compute the same shared key;* i.e., DHKE works.

[1] Although in fact, some workable public key crypto had been invented earlier within the US/UK intelligence community, and not shared with the public. Since quite early in the Cold War, there have been mathematical theorems and proofs held in secret by large governments.

PROOF. There's very little to check here: using the notation of the definition and, in the very middle, the commutativity of multiplication, we have

$$B^\alpha = \left(r^\beta\right)^\alpha = r^{\beta\cdot\alpha} = r^{\alpha\cdot\beta} = \left(r^\alpha\right)^\beta = A^\beta \ .$$

The left end of this equality is the S that Alice computes, while the right end is the one that Bob computes, and they are the same. □

EXAMPLE 5.5.3. Alice and Bob agree in open, public discussion to use the prime $p = 617$ and its primitive root $r = 17$.

Alice privately chooses her secret value $\alpha = 19$ and sends the value

$$A = 17^{19} \equiv 385 \pmod{617}$$

to Bob by insecure e-mail. (All unencrypted e-mail is insecure, of course.)

Bob chooses his secret value $\beta = 13$ and sends the value

$$B = 17^{13} \equiv 227 \pmod{617}$$

to Alice by insecure e-mail.

Alice computes the shared secret

$$S = B^\alpha = 227^{19} \equiv 127 \pmod{617} \ .$$

Bob computes the same shared secret instead by

$$S = A^\beta = 385^{13} \equiv 127 \pmod{617} \ .$$

They then can use this value to do symmetric crypto for the rest of this communication.

What about the practicality of DHKE? As was discussed in §4.4, finding a the [large] prime p (we continue using the notation in Definition 5.5.1) is computationally feasible, as are the several modular exponentiations in the DHKE protocol. There remains the question of finding the primitive root r.

One way is to avoid searching for r more than once. After one prime p and an associated primitive root r for p are found, each user can choose their secret (Alice's α and Bob's β), without overlap or conflict. This is the approach suggested in some Internet standards, see, *e.g.,* [**HC**] and [**LK08**].

Another, more mathematical, strategy is based on the following definition, which we give because of its independent mathematical interest and the amazing life of its namesake.

DEFINITION 5.5.4. A prime p with the property that $2p + 1$ is also a prime is called a **Sophie Germain prime**.

It is not known how many Sophie Germain primes there are, although it is conjectured that there are an infinite number. In fact, there are precise conjectures on the asymptotic density of such primes, as well as algorithmic techniques to generate them efficiently; see [**Sho09**].

EXAMPLE 5.5.5. The first seventeen Sophie Germain primes are

$$2,\ 3,\ 5,\ 11,\ 23,\ 29,\ 41,\ 53,\ 83,\ 89,\ 113,\ 131,\ 173,\ 179,\ 191,\ 233,\ 239$$

The sequence of all Sophie Germain primes is sequence *A005384* at the *On-Line Encyclopedia of Integer Sequences*, `oeis.org`.

The largest known Sophie Germain prime, at the time of this writing, is

$$18543637900515 \times 2^{666667} - 1$$

discovered in 2012 by Philipp Bliedung and a large distributed network of computers.

The reader is asked to investigate the usefulness of Sophie Germain primes in DHKE is explored in the exercises, below.

As mentioned in the last section 5.4, DHKE depends for its security on modular exponentiation be a one-way function: feasible to compute forwards (by fast modular exponentiation) but infeasible to computer backwards (which is an index).

If discrete logs could be computed by a feasible algorithm, then Eve could completely break the security of DHKE. Starting with the public values of A and B, she would compute $\alpha = \mathrm{ind}_r(A)$ and $\beta = \mathrm{ind}_r(B)$. She could then compute S as either B^α, A^β, or, directly, as $r^{\alpha\cdot\beta}$. Having S, she could decrypt all of Alice and Bob's communications as they go by.

Actually, it is not necessary for Eve to be able to compute discrete logs, as long as she can solve a specific computation problem.

DEFINITION 5.5.6. The **Diffie-Hellman problem** [**DHP**] consists of the following question: Given

- a prime p,
- a primitive root $r \in (\mathbb{Z}/p\mathbb{Z})^*$, and
- two elements $a, b \in (\mathbb{Z}/p\mathbb{Z})^*$ which are known to be of the form $a = r^x$ and $b = r^y$ for $x, y \in \mathbb{N}$, although the x and y are not known

compute

- r^{xy} .

An efficient way to compute discrete logs would of course result in a solution of the Diffie-Hellman problem, but it is not known if there might be a solution of the DHP which is does not consist of a full-blown algorithm to compute discrete logs. Since this question is open as of this writing, it makes sense to make the most precise statement: breaking DHKE amounts of solving the DHP, which is not feasible on a classical computer[2]

DHKE is one of the most widely used cryptologic protocols on the Internet. It is part of SSL, TLS, SSH, IPsec, and many VPNs, among others.

[2]As is the case with factoring, there is a known algorithm for a *quantum computer* which solves the DHP efficiently, see [**Sho94**].

Exercises for §5.5.

EXERCISE 5.5.1. Suppose you had an efficient algorithm to generate large Sophie Germain primes. Describe how you would use this to do the initial choice of public parameters for DHKE – go through all of the set-up stages of this protocol, and explain how each can be done *feasibly*, based on either past discussions of calculations we can do feasibly or on new ideas you develop here.

EXERCISE 5.5.2. The number $p = 11717$ is prime, and $r = 103$ is a primitive root of this p. Playing the role of Alice, your instructor computed $A = 5123$.

Please play the role of Bob and do what is necessary to establish a shared key with your instructor: you compute S as in DHKE, and send your B by e-mail so that your instructor can also compute S. Then await further instructions by return e-mail, encrypted with the shared secret.

You may need to perform calculations that are hard to do on a hand calculator. If you have access to, and are familiar with, some computer system like `Octave, Matlab`, or `Mathematica`, you should be able to do the calculations that way. Otherwise, you might try using your favorite search engine on the phrase "fast modular exponentiation applet", or using `wolframalpha.com`.

EXERCISE 5.5.3. Spell out in precise, mathematical detail all of the steps which would be used in a man-in-the-middle attack against DHKE.

5.6. The ElGamal Cryptosystem

As mentioned in the previous section, exponentiation in mod p, where p is a prime known to the public, is a good candidate one-way function: It is fast (feasible) in the forward direction; its inverse, being discrete log with respect to a primitive root r mod p is thought to be infeasible – even when that root is known to the public. This was behind the security of DHKE, and now we discuss how to use this one-way function to set up a more usual public-key cryptosystem: the ElGamal Cryptosystem.

The RSA public-key cryptosystem did its actual encryption by exponentiation in mod $n = pq$. The decryption then was by exponentiation with the multiplicative inverse in mod $\phi(n)$ of the encryption exponent. The owner of the private key – $(p,\ q)$ – knows also the decryption exponent, entirely because $\phi(n)$ can be computed.

Similarly, ElGamal uses an elementary arithmetic operation – multiplication of a numerical form of the message by a random number in some mod – to do the scrambling needed for encryption. Enough information is also passed along in the ciphertext so that the intended recipient, who knows the value of a certain discrete log, can cancel out this scrambling multiplication. Here are the details:

DEFINITION 5.6.1. To start, Alice picks a large prime p, a primitive root r mod p, and a secret value $\alpha \in \mathbb{N}$ satisfying $2 \leq \alpha \leq p-1$. She computes the value $a = r^\alpha$ and then posts her **ElGamal public [encryption] key** (p, r, a) on her website.

Alice's **ElGamal private [decryption] key** is (p, r, α). The association of decryption to encryption keys is by $\mathcal{E} : (p, r, \alpha) \mapsto (p, r, r^\alpha)$.

The message space is $\mathcal{M} = \{m \in \mathbb{Z} \mid 2 \leq m \leq p-1\}$, which we will assume can be interpreted as meaningful messages encoded numerically by some widely known scheme.

Say Bob wishes to send Alice the cleartext $m \in \mathcal{M}$. For each new such message m, he generates a random number $\beta \in \mathbb{N}$ such that $2 \leq \beta \leq p-2$ and builds the ciphertext for **ElGamal encryption** as the two pieces

$$c = e_{(p,r,a)}(m) = (r^\beta \pmod p,\ m \cdot a^\beta \pmod p)\ .$$

When Alice gets the ciphertext $c = (c_1, c_2)$, she can recover the cleartext by **ElGamal decryption**

$$d_{(p,r,\alpha)}(c_1, c_2) = c_2 \cdot c_1^{p-1-\alpha}\ .$$

All of the above parts together form the **ElGamal cryptosystem**.

We first need to know this is correct, in the sense that

PROPOSITION 5.6.2. *With the notation as above in Definition 5.6.1 we have*

$$d_{(p,r,\alpha)}(e_{(p,r,a)}(m)) = m \quad \forall m \in \mathcal{M}\ .$$

PROOF. Just compute:

$$\begin{aligned} d_{(p,r,\alpha)}(e_{(p,r,a)}(m)) &\equiv m \cdot a^{\beta} \pmod{p} \cdot (r^{\beta})^{p-1-\alpha} \pmod{p} \\ &\equiv m \cdot (r^{\alpha})^{\beta} \cdot (r^{\beta})^{p-1-\alpha} \pmod{p} \\ &\equiv m \cdot r^{\alpha\beta+\beta(p-1)-\alpha\beta} \pmod{p} \\ &\equiv m \cdot (r^{p-1})^{\beta} \pmod{p} \\ &\equiv m \cdot 1^{\beta} \pmod{p} \\ &\equiv m \pmod{p} \end{aligned}$$

Note that the power $p-1-\alpha$ as the exponent of the c_1 term in decryption is to make $(c_1^{-1})^{\alpha}$ without using negative powers, by applying Theorem 5.1.2. □

Graphically:

ElGamal cryptosystem:

Alice	on public network	Bob
pick a prime p find a primitive root r choose $\alpha \mid 2 \leq \alpha \leq p-1$ compute $a = r^{\alpha} \pmod{p}$ publish public key	$\rightarrowtail (p, r, a)$	download public key
		given message $m \in \mathcal{M}$ (so $2 \leq m \leq p-1$) choose $\beta \mid 2 \leq \beta \leq p-2$ compute $c_1 = r^{\beta} \pmod{p}$ and $c_2 = m \cdot a^{\beta} \pmod{p}$
receive ciphertext	$(c_1, c_2) \leftarrowtail$	transmit ciphertext
compute cleartext $m = c_2 \cdot c_1^{p-1-\alpha}$		

There is a nice digital signature algorithm associated with ElGamal:

DEFINITION 5.6.3. Suppose Alice has ElGamal private key (p, r, α) and wishes to digitally sign the message $m \in \mathcal{M}$. She first chooses a random $\gamma \in \mathbb{N}$ satisfying $1 < \gamma < p-1$ and $\gcd(\gamma, p-1) = 1$.

The digital signature on m is $(r^{\gamma} \pmod{p}, \gamma^{-1} \cdot (m - \alpha r^{\gamma}) \pmod{p-1})$, where the inverse is taken in mod $p-1$.

To verify the signature (x, y) on message m using Alice's public key (p, r, a), Bob checks to see if $a^x x^y \equiv r^m \pmod{p}$: if so, he accepts; if not, he rejects.

Again, we would like to know this does the right thing:

PROPOSITION 5.6.4. *Using the notation as above, Bob will accept all signed messages produced by Alice.*

PROOF. Assuming the signed message (m, x, y) was produced by Alice as above, we compute:

$$\begin{aligned} a^x x^y &\equiv a^{r^\gamma}(r^\gamma)^{\gamma^{-1}(m-\alpha r^\gamma)} \pmod{p} \\ &\equiv (r^\alpha)^{r^\gamma}(r^\gamma)^{\gamma^{-1}(m-\alpha r^\gamma)} \pmod{p} \\ &\equiv r^{\alpha r^\gamma + \gamma\gamma^{-1}(m-\alpha r^\gamma)} \pmod{p} \\ &\equiv r^{\alpha r^\gamma + m - \alpha r^\gamma} \pmod{p} \\ &\equiv r^m \pmod{p} \end{aligned}$$

So Bob will accept. □

Graphically:

ElGamal digital signatures:

Alice	on public network	Bob
find prime p and primitive root r choose $\alpha \mid 2 \le \alpha \le p-1$ compute $a = r^\alpha \pmod{p}$ publish public key	$\rightarrowtail (p, r, a)$	download public key
given message $m \in \mathcal{M}$ (so $2 \le m \le p-1$) pick random γ s.t. $1 < \gamma < p-1$ and $\gcd(\gamma, p-1) = 1$ compute $s_1 = r^\gamma \pmod{p}$ and $s_2 = \gamma^{-1} \cdot (m - \alpha r^\gamma) \pmod{p-1}$ transmit signed message	$\rightarrowtail (m, s_1, s_2)$	receive signed message
		if $a^{s_1} \cdot s_1^{s_2} \equiv r^m \pmod{p}$ ACCEPT otherwise, REJECT

Exercises for §5.6.

EXERCISE 5.6.1. You instructor still likes the prime $p = 11717$ with primitive root $r = 103$ from an earlier exercise 5.5.2 on DHKE. In addition, your instructor has calculated the value $a = 1020$ to complete an ElGamal public key $(p, r, a) = (11717, 103, 1020)$.

Using this public key, you want to send a message to your instructor, which should consist of the number 42 (it is, after all, the answer to "life, the universe, and everything"). What ciphertext will you send? Show your work!

EXERCISE 5.6.2. Now your instructor wants to send you your grade on a recent test by e-mail, and to prove that this e-mail does in fact originate with your instructor, the email contains both the score value of 97 and the addendum "This score value signed with an ElGamal Digital signature using my public key [the same instructor's public key as above in exercise 5.6.1, being $(p, r, a) = (11717, 103, 1020)$]; the signature has the value $(6220, 10407)$."

Do you accept this as truly coming from your instructor? Show your work!

EXERCISE 5.6.3. Create an ElGamal public key and e-mail it to your instructor. Wait for a reply message which is ElGamal encrypted, then mail the cleartext back to your instructor.

Also, use your public key to sign the number (=message) 17. Send the signed number to your instructor and wait to hear if the signature is accepted or not.

Bibliography

[AB09] Sanjeev Arora and Boaz Barak, *Computational Complexity: A Modern Approach*, Cambridge University Press, 2009.

[AKS04] Manindra Agrawal, Neeraj Kayal, and Nitin Saxena, *Primes is in p*, Annals of mathematics (2004), 781–793.

[Bou04] Nicolas Bourbaki, *Theory of sets*, Springer, 2004.

[DH76] Whitfield Diffie and Martin E Hellman, *New directions in cryptography*, Information Theory, IEEE Transactions on **22** (1976), no. 6, 644–654.

[FS03] Niels Ferguson and Bruce Schneier, *Practical cryptography*, vol. 23, Wiley New York, 2003.

[Gau86] Carl Friedrich Gauß, *Disquisitiones Arithmeticae, 1801. English translation by Arthur A. Clarke*, 1986.

[Har05] Godfrey Harold Hardy, *A Mathematician's Apology*, 2005, First electronic edition, available at `http://www.math.ualberta.ca/mss`.

[HC] Dan Harkins and Dave Carrel, *RFC 2409: The Internet Key Exchange (IKE), November 1998*, Status: Proposed Standard.

[HW79] Godfrey Harold Hardy and Edward Maitland Wright, *An introduction to the theory of numbers*, Oxford University Press, 1979.

[LK08] M Lepinski and S Kent, *RFC 5114-Additional Diffie-Hellman Groups for Use with IETF Standards*, 2008.

[Lub96] Michael George Luby, *Pseudorandomness and Cryptographic Applications*, Princeton University Press, 1996.

[MVOV96] Alfred J Menezes, Paul C Van Oorschot, and Scott A Vanstone, *Handbook of applied cryptography*, CRC press, 1996.

[NC10] Michael A Nielsen and Isaac L Chuang, *Quantum computation and quantum information*, Cambridge university press, 2010.

[PS04] Jonathan A Poritz and Morton Swimmer, *Hash woes*, Virus Bulletin (2004), 14–16.

[RSA78] Ronald L Rivest, Adi Shamir, and Len Adleman, *A method for obtaining digital signatures and public-key cryptosystems*, Communications of the ACM **21** (1978), no. 2, 120–126.

[Sha48] C. E. Shannon, *A mathematical theory of communication*, Bell Systems Technical Journal **27** (1948), 379–423, 623–656.

[Sha49] ———, *Communication theory of secrecy systems*, Bell System Technical Journal **28** (1949), no. 4, 656–715.

[Sho94] Peter W Shor, *Algorithms for quantum computation: discrete logarithms and factoring*, Foundations of Computer Science, 1994 Proceedings., 35th Annual Symposium on, IEEE, 1994, pp. 124–134.

[Sho09] Victor Shoup, *A computational introduction to number theory and algebra*, Cambridge University Press, 2009, on-line at `http://shoup.net/ntb/`.

[Sta02] Richard Stallman, *Free Software, Free Society: Selected Essays of Richard M. Stallman*, Lulu.com, 2002.

[WY05] Xiaoyun Wang and Hongbo Yu, *How to break md5 and other hash functions*, Advances in Cryptology–EUROCRYPT 2005, Springer, 2005, pp. 19–35.

Index