

Secrets... and lies

How can we transmit information in such a way that only authorised persons can understand it? How can we be sure that the information we transmit reaches its destination without being altered? Moreover, how can we be sure of the origin of a message, and so trust its content? In this chapter we shall deal with these problems. We shall first examine the earliest *classic* cryptographic methods, rapidly outlining their development along the centuries, and then we shall discuss the most recent research about *public key* cryptography. For further details about the history and development of cryptography, the reader can have a look at the good popular scientific book [58].

7.1 The classic ciphers

Humanity has always felt the need for efficient methods to communicate in a secret and secure way: by this we mean the ability of sending messages that can be easily read by the addressees and cannot possibly be deciphered by unauthorised people. This millennium-old problem is extremely important today, when the advances in electronic communication systems make exchanging information both easier and more vulnerable.

The earliest examples of secret messages appear in the *Histories* by Herodotus, the Greek historian who lived in the 5th century BCE and chronicled the contemporary Greco-Persian wars.

7.1.1 The earliest secret messages in history

Herodotus was an extraordinary narrator: he had an unbelievable talent for reporting what he had seen and been told in his travels in Asia Minor, Greece, Africa, Sicily and so on.

In Book VII of the *Histories* he tells how Xerxes, having succeeded his father Darius, after crushing a rebellion in Egypt, is about to wage war on Greece: preparations for the expedition are made by creating a formidable army. The pages narrating this

preparations are engrossing, with a survey of the army, a detailed portrayal of the costumes and the armours of each of the peoples composing the Persian army, the description of the fleet. Finally, the expedition leaves: but somebody has warned the Greeks of Xerxes's actions and Book VII ends with following passage:

The Lacedemonians [= Spartans] had been informed before all others that the king was preparing an expedition against Hellas; and thus it happened that they sent to the Oracle at Delphi, where that reply was given them which I reported shortly before this. And they got this information in a strange manner; for Demaratos the son of Ariston after he had fled for refuge to the Medes was not friendly to the Lacedemonians, as I am of opinion and as likelihood suggests supporting my opinion; but it is open to any man to make conjecture whether he did this thing which follows in a friendly spirit or in malicious triumph over them. When Xerxes had resolved to make a campaign against Hellas, Demaratos, being in Susa and having been informed of this, had a desire to report it to the Lacedemonians. Now in no other way was he able to signify it, for there was danger that he should be discovered, but he contrived thus, that is to say, he took a folding tablet and scraped off the wax which was upon it, and then he wrote the design of the king upon the wood of the tablet, and having done so he melted the wax and poured it over the writing, so that the tablet (being carried without writing upon it) might not cause any trouble to be given by the keepers of the road. Then when it had arrived at Lacedemon, the Lacedemonians were not able to make conjecture of the matter; until at last, as I am informed, Gorgo, the daughter of Cleomenes and wife of Leonidas, suggested a plan of which she had herself thought, bidding them scrape the wax and they would find writing upon the wood; and doing as she said they found the writing and read it, and after that they sent notice to the other Hellenes. These things are said to have come to pass in this manner. (Translation by G.C. Macaulay.)

Moreover, it is well known that in the Battle of Salamis the Greeks, having been informed of Xerxes's expedition thanks to this stratagem and so being ready to confront him, managed in 480 BCE to defeat the Persians. So a secret, cleverly hidden message, changed the outcome of a war.

Herodotus himself, in Book V of the *Histories*, tells the story of Histiaios who desiring to signify to Aristagoras that he should revolt, was not able to do it safely in any other way, because the roads were guarded, but shaved off the hair of the most faithful of his slaves, and having marked his head by pricking it, waited till the hair had grown again; and as soon as it was grown, he sent him away to Miletos, giving him no other charge but this, namely that when he should have arrived at Miletos he should bid Aristagoras shave his hair and look at his head: and the marks, as I have said before, signified revolt. This thing Histiaios was doing, because he was greatly vexed by being detained at Susa. He had great hopes then that if a revolt occurred he would be let go to the sea-coast; but if no change was made at Miletos he had no expectation of ever returning thither again. (Translation by G.C. Macaulay.)

It is clear that as soon as the enemy suspects the existence of a hidden message, the obvious countermove is to inspect with the greatest care all possible hiding places. In the episodes told by Herodotus, the suspect would be searched until the message hidden under the hair is found, or the tablet would be examined meticulously until the place where the message was written is spotted.

We conclude this historical preamble with a last anecdote (see [58]).

Mary Stuart, Queen of Scots, imprisoned in 1568 by Queen Elizabeth, was a prisoner for 18 years. In 1586 a plot to free her and simultaneously assassinate Queen Elizabeth was organised: the conspirators deemed it necessary for their plan to be approved by the Queen of Scots. To do so, they used hidden and ciphered secret messages. But both the presence of a double-crosser and the deluded certainty of being able to write freely in the messages, in the (mistaken) confidence in the cryptosystem they used being indecipherable, drove Mary to write more than she should have; this gave Queen Elizabeth the proof of her involvement in the plot and led to her death sentence.

These examples and many more show how, mainly during wartime, the need for devices to send messages in such a way that the adversaries could not discover them has been felt for centuries. The most spontaneous way is to hide the message as the above episodes relate: this technique is called *steganography*.

Another way to send a message in such a way that the enemy cannot understand it is obtained by hiding *not the message, but its meaning*. In this case we are dealing with *cryptography*. We are *enciphering* a message so that it can be read by whomever obtains it, but only the actual addressee is able to decipher it, while the enemy cannot, even if he gets hold of it. A first, simple example of ciphering of a message consisted in substituting Greek characters for Latin ones. But perhaps one of the first recorded examples of a ciphered message in the history dates back to Julius Caesar. Thanks to Suetonius's *On the Life of the Caesars* (2nd century CE), we know one of the systems used by Caesar to encipher his messages: he shifted by three positions, with respect to its position in the alphabet, each letter of the message to be sent.

If we denote by lower case letters the 26 letters of the alphabet, each letter of the message (*plaintext*) will be substituted with the letter following it by three positions, which we shall write in upper case: so we get a new message (*ciphertext*). The explicit correspondence between the letters is described in Table 7.1.

The *enciphering* or *encryption* is the rule describing how to pass from one alphabet to the other one, that is, allowing us to rewrite a message so to make it unreadable for those who do not know the rule. For instance, if the message to be sent is

attack tomorrow	(plaintext),
-----------------	--------------

the result after enciphering is

DWDFN WRPRUURZ	(ciphertext).
----------------	---------------

Table 7.1. Cipher used by Caesar

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

A system like this, in which the cipher alphabet is obtained from the plain alphabet by moving each letter by a fixed number of positions, is called *Caesar cipher*. In English there are altogether 26 possible Caesar ciphers, or rather 25, as clearly if a letter is moved by 26 positions it comes back to its starting point and the ciphered message is equal to the original one. In other words, we may define a bijection between the possible cipher alphabets and the residue classes modulo 26, that is with the integers n such that $0 \leq n \leq 25$. Given such an integer n , called *key*, the corresponding cipher alphabet is the one moving the letters of the plaintext by n positions, that is to say, the alphabet obtained by an n -position *shift*. Clearly the value $n = 0$ corresponds to the initial alphabet, that is to the plaintext.

If a message that has possibly been enciphered using a Caesar cipher has been intercepted, it suffices, in order to decrypt it, to use the 26, or rather the 25, keys of the possible cipher alphabets. So this enciphering can be sidestepped very easily, especially so if one has good computing instruments, as we have today, while Caesar and his enemies had not.

Refining this principle, we may use as enciphering, rather than just the *shifts*, all possible permutations of the 26 letters. In practice, each permutation of the set $\{0, 1, 2, \dots, 25\}$, called *key* as above, determines a cipher alphabet, and the other way around: for instance, for the identity permutation it suffices to have 0 correspond to *A*, 1 to *B*, 2 to *C* and so forth.

But how can we remember the key? We should remember the whole letter sequence, lacking a specific scheme to memorise it. However, there is a good system to generate a permutation of the alphabet that can be easily memorised: it consists in using a key that is itself determined by a *key word* or a *key phrase*, or any letter string we can easily remember. Let us see an example to clarify this method.

Example 7.1.1. Assume we have chosen as key phrase the following:

to be or not to be that is the question.

First of all, remove the spaces between the words of the key phrase and then the repetitions, obtaining in our case

tobernhaissqu.

The cipher alphabet will be constructed by putting in the order, under the plain alphabet, first the letters of the key word modified as above, and then the letters of the plain alphabet not appearing in the key phrase, in the usual alphabetic order. So we get:

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
T	O	B	E	R	N	H	A	I	S	Q	U	C	D	F	G	J	K	L	M	P	V	W	X	Y	Z

In this way we have associated to the key, that is to the phrase *to be or not to be that is the question*, the cipher alphabet shown. We may verify that the permutation determined by the key word and that determines the alphabet is described by the following table:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
19	14	1	4	17	13	7	0	8	18	16	20	2	3	5	6	9	10	11	12	15	21	22	23	24	25

So, if the message to send were

it is the early bird that gets the worm

the result after being enciphered in this way would become

IM IL MAR RTKUY OIKE MATM HRML MAR WFKC

In general, in this way the number of possible keys, and so of cipher alphabets, increases quickly from 26 (Caesar ciphers) to $26!$, the number of all possible permutations of 26 elements (see Exercise A1.11). This number is

51090942171709440000,

that is, about $51 \cdot 10^{18}$, or more than *fifty billion billion*: if an adversary intercepts the message and suspects this enciphering method has been used he cannot possibly *try to decrypt it by trial and error*. To realise the hugeness of this number it suffices to recall that the Big Bang occurred approximately 15 billion years ago. So whoever has to *send secret messages* may rest easy and relax: nobody can possibly decrypt them! But are things really like this?

Unfortunately, the answer is no: the frequency with which a given letter appears in a text long enough, and other factors depending on the alphabet used can reduce substantially the number of attempts necessary to find the key! We shall return on this in next section.

As regards our ciphers, an enciphering using a single cipher alphabet, as those seen so far, is called *monoalphabetic cipher*. However, we may consider using more than one cipher alphabet. How?

Suppose we want to use $s \in \mathbb{N} \setminus \{0\}$ cipher alphabets. Then, divide the message into s -letter blocks and successively encipher the letters in each block with the s alphabets, always using them in the same order. In other words, denoting by A_i , $1 \leq i \leq s$, the s alphabets, all the letters that are in the i th position of a block will be enciphered with the same alphabet A_i . Such a cipher is called *periodic polyalphabetic cipher*. If s is equal or greater than the length of the message we shall simply have what is called an (*aperiodic*) *polyalphabetic cipher*.

The first example of this kind is apparently due to Leon Battista Alberti, in the second half of 15th century: he proposed the use of *two* cipher alphabets for each message. His idea was improved later by Vigenère in the second half of 16th century. Vigenère proposed that each message should be enciphered using 26 cipher alphabets. The 26 cipher alphabets are shown in table 7.2, where the integer appearing in each line is exactly the shift key giving the cipher alphabet.

To fix ideas, and prepare the mathematical model, we may label the letters of the message using integers as described by table 7.3.

The integers from 0 to 25 are called *numerical equivalents* of the alphabet letters. In this way, as already remarked, we write 0 in the place of a , 12 in

Table 7.2. Vigenère table

	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
0	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
2	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
3	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
4	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
5	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
6	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
7	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
8	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
9	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
10	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
11	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
12	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
13	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
14	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
15	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
16	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
17	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
18	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
19	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
20	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
21	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
22	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
23	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
24	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
25	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

the place of m , and so forth; if necessary, we shall use the notation $a \leftrightarrow 0$, $m \leftrightarrow 12$. We might further use other numbers to denote spaces in the text, commas, diacritics and every other symbol that might be useful to reconstruct more easily the text. For simplicity, we shall not use these signs.

Table 7.3. Numerical equivalents of the 26 letters

a \longrightarrow 0	h \longrightarrow 7	o \longrightarrow 14	u \longrightarrow 20
b \longrightarrow 1	i \longrightarrow 8	p \longrightarrow 15	v \longrightarrow 21
c \longrightarrow 2	j \longrightarrow 9	q \longrightarrow 16	w \longrightarrow 22
d \longrightarrow 3	k \longrightarrow 10	r \longrightarrow 17	x \longrightarrow 23
e \longrightarrow 4	l \longrightarrow 11	s \longrightarrow 18	y \longrightarrow 24
f \longrightarrow 5	m \longrightarrow 12	t \longrightarrow 19	z \longrightarrow 25
g \longrightarrow 6	n \longrightarrow 13		

How may we remember the sequence of the s alphabets to be used in enciphering our message? *By memorising it using a key word.* We may use a word whose length s represents the period by which the alphabets are repeated.

Example 7.1.2. Assume we have chosen the key word **FISH** and we want to encipher the sentence “**shoot now**”.

The rules to be followed in the encipher are contained in the key word we have chosen, in the sense that we shall use as cipher alphabets, repeating each in each 4-letter word (**shoo** | **tnow**), the lines of Vigenère table corresponding successively to the letters F, I, S, H . In our example, the lines are the ones numbered 5, 8, 18, and 7. Then the first letter of the message, the s , shall be enciphered with the letter that is in the position of s using as cipher alphabet that of line 5, corresponding to the letter F of the key word, up to the second letter o that shall be enciphered with the line corresponding to the letter H ; then we start again, using the alphabet corresponding to the letter F for the letter t , and so on.

We conclude this section with another anecdote (see *Scientific American*, August 1977).

In 1839, Edgar Allan Poe, from the pages of a Philadelphia periodical, asked his readers for cryptograms with monoalphabetic substitutions, and guaranteed he would solve them. Among many other ones, he received the following handwritten cryptogram:

GE JEASGD XV,
ZIJ GL MW LAAM XZY ZMLWHFZEK EJLVDXW KWKE TX LBR ATGH LBMX AANU BAI
VSMUKKSS PWNVLWK AGH GNUMK WDLNZWEG JNBXVV OAEG ENWBZWMGY MO MLW WNBX MW
AL PNFD CFPKH WZKEH HSSF XKIYAHUL? MK NUM YEXDM WBXY SBC HV WYX PHWKG NAMCUK?

The letters in bold correspond to upper case letters.

Having read the message, Poe replied that it consisted of random symbols, not corresponding to any monoalphabetic substitution. More than one hundred years later, in 1975, the mathematician Bryan J. Winkel, and the research chemist Mark Lyster, who took part in the course in cryptography given by Winkel, decrypted the message. In fact, it was not a monoalphabetic substitution, but did not even consist of random letters. Moreover, there were some errors probably due to the transcription of the text (see Exercise B7.20). The solution is as follows:

Mr. Alexander,

How is it that the messenger arrives here at the same time with the Saturday courier and other Saturday papers when according to the date it is published three days previous? Is the fault with you or the postmasters?

7.2 The analysis of the ciphertext

We have just described some techniques to encipher messages that look to be unbreakable, at least if our only way is to proceed by trial and error and we are not incredibly lucky. Is this all, or is there some other information we can use?

Let us look at things from the point of view of the adversary, who wants to decrypt the message at any cost. We know the message has been enciphered using a monoalphabetic substitution and, as we have remarked, we cannot proceed by trial and error, if we are to solve the problem in an admissible time.

So we have to use other methods, independent of the kind of key that has been used, and so of the cipher alphabet it determines. How may we proceed, having only a page of ciphertext? We subject it to a *text analysis*: depending on the language the text is written in, we take into account its properties. In a language like Italian, where most words end with a vowel, most of the symbols at the end of the words of the ciphertext will be vowels. More in general, much information is given by *frequency analysis*. In each language some letters appear with greater frequency, some more rarely. Linguistic and statistical studies have found the frequency of the 26 letters of English alphabet:

Letter	%	Letter	%	Letter	%	Letter	%
a	7,3	h	3,5	o	7,4	u	2,7
b	0,9	i	7,4	p	2,7	v	1,3
c	3,0	j	0,2	q	0,3	w	1,6
d	4,4	k	0,3	r	7,7	x	0,5
e	13,0	l	3,5	s	6,3	y	1,9
f	2,8	m	2,5	t	9,3	z	0,1
g	1,6	n	7,8				

So if we know that the text to decrypt is written in English, and if it is long enough, we may determine the frequencies of the letters in the text. Those appearing with the greatest frequency might possibly be *es*, or *ts*. Then we may look for correspondences by trial and error, examining successively the letters with smaller frequency, looking for pieces of the puzzle falling into place. If we get meaningless words, we adjust our tentative assignments. Further information is given by the frequency of double letters, the greater or smaller likelihood that certain letters are found close together, and so on.

Example 7.2.1. We get back to the example we saw in the previous section, to see how to use efficiently the techniques just described to decrypt the enciphered message

IM IL MAR RTKUY OIKE MATM HRML MAR WFKC

We write down the frequency of the letters in the message:

Letter	Times	Letter	Times	Letter	Times
A	3	J	0	S	0
B	0	K	3	T	2
C	1	L	2	U	1
D	0	M	6	V	0
E	1	N	0	W	1
F	1	O	1	X	0
G	0	P	0	Y	1
H	1	Q	0	Z	0
I	3	R	4		

As this is a short message, frequency analysis might be misleading, but in any case we may try to use it. We might also exploit the fact that certain letters are more likely to end a word, or the length of the words, but this attempts are easily foiled by breaking up the message in blocks of the same length, which makes it difficult to reconstruct the single words. However, at this stage we shall use everything we know. So let us analyse our text. The most frequent letters, *M* and *R*, might correspond to *e* and *t*. Moreover, the repeating subsequence *MAR* is likely to be the article *the*. So we may tentatively try the correspondences

$$M = t, \quad A = h, \quad R = e,$$

and for the next most frequent letters of the ciphertext, *I* and *K*, we might try the next most frequent alphabet letters *n* and *r*. In this way we obtain:

nt nL the eTrUY OIrE thTt HetL the WFrC.

Clearly, the choice $I = n$ is not promising, and anyway, if the other choices are right, it looks like *I* must represent a vowel, or the sentence would begin with the “word” *nt*. Trying out *o*, *i* and *a*, the best choice appears to be *i*. So we get:

it iL the eTrUY OIrE thTt HetL the WFrC

and after some more attempts and frequency analyses we get the solution.

It is clear that this example is not too typical because we are analysing a very short text, but the important thing is to emphasise the fact that the far too many theoretical possibilities to be considered in order to find the key decrease enormously by using data about the language and making some educated guesses and backtracking.

Analysing a Caesar cipher by studying frequencies is clearly even easier.

Vigenère system too can be attacked with a suitably adapted frequency analysis, if its period is known. For instance, if we assume that the period, that is to say the length of the key word is four, then we have to line up the letters of the ciphertext in four columns, as follows:

```

* * * *
* * * *
* * * *
.....
.....

```

If in the same column there are equal cipher letters, they represent the same letter of the plaintext, as all the letters in the same column are enciphered with the same cipher alphabet.

Coming back to Example 7.1.2, the pattern is

$$\begin{array}{ccc} \text{shoo} & \longrightarrow & \text{XPGV} \longrightarrow 23\ 15\ 6\ 21 \\ \text{tnow} & \longrightarrow & \text{YVGD} \longrightarrow 24\ 21\ 6\ 3 \end{array}$$

and, as is immediately seen, the letter G (or its numerical equivalent 6) appearing in the third column is repeated and corresponds to the same letter o of the plaintext.

These remarks allow us, with due caution, to use frequency analysis *on each column separately* to reconstruct the key word. Moreover, a German cryptologist who lived at the end of 19th century, F. W. Kasiski, found a method to determine the period of the alphabet, and this partly explains the loss of interest for this kind of ciphers.

Remark 7.2.2. Some of these methods can also be applied to ancient inscriptions: clearly the people writing them did not, in general, intend to encipher a message, but for us those texts actually are enciphered messages we have to decrypt. Decrypting an unknown form of writing is something of a magic, as it allows to enter a past world, to get to know a dead civilisation, to call to mind a remote age. The main example are Egyptian hieroglyphics: the most ancient ones date back to fourth millennium BCE. Interest in them was aroused in 16th century when Pope Sixtus V decreed that a new road network should be built in Rome, putting at the crossroads some Egyptian obelisks: confronted with those puzzles, many tried to understand their meaning. The most famous archaeological find with hieroglyphics is undoubtedly the *Rosetta stone*, made of black basalt, discovered in 1799 near Nile's delta and engraved in 196 BCE: it is an inscription regarding a decree by an assembly of priests honouring the Pharaoh and, as is well known, it carries the same text in three versions: hieroglyphic Egyptian, Demotic Egyptian and Greek. The Greek text was easily translated and so it became, in a sense, the *plaintext* against which the two other texts could be compared: it yielded both a great opportunity and an irresistible challenge. J.F. Champollion measured himself against it and, in 1822, solved the mystery. Today the stone is kept in the British Museum in London.

Archaeologists have deciphered several ancient writing systems and languages, but many of them are still undeciphered, among them Etruscan. The most intriguing deciphering has perhaps been that of *Linear B*, a Mediterranean script, dating back to Bronze Age. The renowned English archaeologist Sir Arthur Evans, during his excavations in Crete in 1900, unearthed a great number of clay tablets bearing writings, partly in a script which was called *Linear B*. These tablets made up the archives of Cretan palaces. They were deciphered in 1953 by M. Ventris and J. Chadwick. It would be very long to relate the whole story of the deciphering: suffice it to say that it could form the plot of a thrilling detective novel. Due to the understanding of *Linear B*, the political and social situation of Cretan society has been now reconstructed, at least in its main lines.

We have said above that the purpose of the authors of the inscriptions was not, in general, to encipher them. We kept on the safe side by saying *in general*: indeed,

some scholars have recently discovered the presence of cryptographical methods in Egyptian hieroglyphics. Apparently, some of them were enciphered by order of the Pharaohs, using several techniques, among which enciphering by substitution.

We are coming to the end of our digression, which teaches us that history is full of messages to be deciphered. Every discovery unearths a secret whose key was hidden.

Notice that, in trying to understand a message enciphered using a monoalphabetic substitution, we have used quite subtle statistical techniques. Apparently, these cryptanalytical techniques have been invented by Arabs during Middle Ages. A subject like this, in fact, can only appear and flourish within a civilisation possessing an advanced knowledge of mathematics, linguistics, and statistics. Arab civilisation undoubtedly had these traits. The most ancient document describing explicitly the frequency method dates back to 9th century, and is due to Abū Yūsuf ibn Ishāq al-Kindī, known as *the Arab Philosopher*, who in his monograph *On Deciphering Cryptographic Messages* described in detail techniques based on statistics and Arabic phonetics and syntax to be used to decrypt documents.

To complete the historical sketch of cryptography we deal in brief with the machines that, along the centuries, have been devised to put in practice various ciphers.

7.2.1 Enciphering machines

The first enciphering machine is the so-called *cipher disc* by Leon Battista Alberti, which is made up of two concentric copper discs, of different diameters, which can rotate one with respect to the other around a central axis. Along the circumferences of the two discs two alphabets are engraved. To encipher a message using Caesar cipher shifting letters by two positions, it suffices to put the *a* of the internal disc, representing the plain alphabet, next to the *C* of the external disc, representing the cipher alphabet (with key $n = 2$). After this simple operation, to encipher the message it suffices, *without any further rotation of the discs*, to read successively the letters on the external disc corresponding to the letters on the internal one. It is a very simple and effective device, which has been in use for several centuries.

The same device can be also used, in quite a natural way, for a polyalphabetic enciphering: changing the position of the second disc means exactly choosing a new alphabet.

As already remarked, this enciphering machine has lived on for several centuries, up to the moment, towards the end of World War I, it has been superseded by the famous *Enigma* machine, invented and constructed by Arthur Scherbius and Richard Ritter, and used until World War II by the German army. In a first form, it consisted of the following three elements, connected by electric wires (see figure 7.1):

- an alphabetic keyboard, to input the plaintext;
- a *scrambler unit*, which is the part actually performing the enciphering;
- a board with as many light bulbs as the letters of the plain alphabet, devised in such a way that the processed electric signal would light the lamp corresponding to the enciphered letter.

The scrambler unit is the system's main part. It is the device actually enciphering the message: it consisted of a thick rubber disc through which a complex network of electric wires passed. For instance, to encipher the letter *a* with the letter *D*, *a* is input on the keyboard: in this way the electric current enters the scrambler, follows the route through the electric wires, and lights the lamp corresponding to the letter *D*.

Later, Scherbius modified the machine, substituting a scrambling rotor for the original scrambler: in this way, the scrambling disc automatically rotated by one twenty-sixth of a revolution (if the alphabet consisted of 26 letters) after enciphering each letter. So, to encipher the next letter, a different cipher alphabet is used. The rotating scrambler defines 26 cipher alphabets. Further improvements substituted the single scrambler with three scrambling rotors, and introduced a *reflector*, which could reflect the signals processed by the rotors, adding complexity to the machine. In short, it was a very sophisticated enciphering machine, so much so that Scherbius believed that *Enigma* generated unbreakable coded messages. In 1943, during World War II, the English used the *Colossus* computers to decrypt messages generated by *Enigma*. It is very interesting to notice that many of the researchers who collaborated to the breaking of Enigma, among which the mathematician A. Turing, perhaps inspired by the peculiarities of the problem, went on to give fundamental contributions to the development of computer science and artificial intelligence.

7.3 Mathematical setting of a cryptosystem

Let us go back to enciphered messages. The science of decrypting messages for which the key is not known is called *cryptanalysis*.



Fig. 7.1. Enigma's keyboard and display

So on the one hand there are the *cryptologists*, designing methods to encipher messages in such a way that they cannot be read by unauthorised people, on the other there are the *cryptanalysts*, who try to decrypt messages, looking for weaknesses in the cryptographic system. The interaction of these two subjects, *cryptology* and *cryptanalysis*, which taken together form *cryptography* and deal, from different viewpoints, with the same object, leads, as can be expected, to ever more complex and secure enciphering systems.

In this section we shall show how to give a proper mathematical layout to cryptology-cryptanalysis.

In table 7.4 we give a short glossary of the terms used in cryptography. We further remark that the root *crypt-* derives from Greek *kryptos*, meaning “hidden, secret”.

Before going on, some remarks are in order.

- The alphabets used for plaintexts and ciphertexts can be different among them and with respect to the one commonly used in the language. In general, it is convenient to write messages using, rather than letters, integer numbers, which are more suitable to the description of the transformations, that is, the enciphering methods, to be used.
- The transformation procedure, that is, the function describing the passage from plaintext to ciphertext, must be *bijective* if we want to be able to reverse the procedure to decipher the message and find back the original text rather than something else. The crucial thing is that the person who shall have to decipher the message has to be in possession of the key!

Table 7.4. Glossary of cryptography

Lexeme	Meaning
Plaintext	Original message to be sent in a secret way, or string of symbols in a given alphabet representing the message or text to be enciphered
Ciphertext	Modified, disguised version of the plaintext
Encipher, (encrypt)	Convert a plaintext into a ciphertext
Decypher, decrypt	Convert a ciphertext into a plaintext
Cipher	<i>Method</i> used to convert a plaintext into a ciphertext
Key	Data determining both a particular enciphering and the corresponding deciphering rule, among all the possible ones: in the first case it is called <i>cipher key</i> , in the second <i>decipher key</i>
Cryptology	Science of enciphering messages
Cryptanalysis	Science of interpreting enciphered messages

- Why do we need a key? Is it not sufficient to deal with the enciphering and deciphering transformations? The fact is, once we have perfected a system to send enciphered messages, changing often the key offers a greater security without having to modify the whole enciphering system. In other words, we could describe the system as a combination lock, and the key as one of several possible combinations. The lock is the enciphering system we are using, while the key is the combination.
- We are not especially interested here in detailing precisely the set of possible keys. The important thing we want to emphasise is that it has to have a size that is not too small, or else it would be feasible for a cryptanalyst to try out all possible keys for that kind of cipher. For instance, Caesar cipher has a far too small number of keys.

All in all,

- *the task of the cryptologist* is to invent systems to transform a plain message into a cipher message; such systems are called *cryptosystems*;
- *the task of the cryptanalyst* is to oppose this activity, finding ways to interpret enciphered messages, in general without the authorisation of the sender.

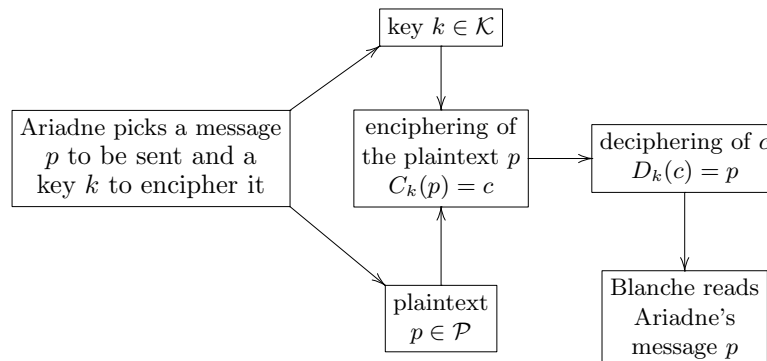
In conclusion, a cryptosystem consists of:

- a set \mathcal{P} , consisting of the possible plaintexts; a single plaintext shall be denoted by the letter p ;
- a set \mathcal{K} , called key set. We shall denote a key by the letter k . Each element $k \in \mathcal{K}$ determines an enciphering transformation C_k and a deciphering transformation D_k , inverse of each other. In particular, $D_k \cdot C_k(p) = p$;
- a set \mathcal{C} consisting of the enciphered messages. We shall denote one of these ciphertexts by the letter c .

So, given a cryptosystem determined by the triple $(\mathcal{P}, \mathcal{C}, \mathcal{K})$ with

$$\mathcal{P} = \{\text{plaintexts}\}, \quad \mathcal{C} = \{\text{ciphertexts}\}, \quad \mathcal{K} = \{\text{keys}\},$$

the communication between two persons, Ariadne and Blanche, is described by the following diagram:



In general the messages, both plain and enciphered ones, are split up into *unitary messages*. A *unitary message* may consist of a *single* letter, or a pair of letters (*digraph*), a triple of letters (*trigraph*), or *s*-letter blocks. The advantage of dividing a message into blocks of a fixed length is in preventing the easy recognition of the beginning and end of the words, making the cryptanalysis based on frequencies more difficult.

Suppose for the time being to have unitary messages consisting of a single letter each, in a given alphabet. To describe mathematically a cryptosystem, the most effective way, as already remarked, is to associate with each symbol of our alphabet an *integer number*. Assume for simplicity the alphabet in which we write our messages to be the English one, and consider its *numerical equivalents* (see Table 7.3 on page 324).

As we are using a 26-letter alphabet, it is natural to perform all mathematical operations on the numerical equivalent of letters modulo 26. In this way 26 is identified with 0, that is with the letter *a*, and so forth. As already suggested, we might use other numbers to denote spaces in the text, commas, diacritics and other symbols which may help in reconstructing more easily the text, but we are not presently interested in them. In general, if the unitary message is an *s*-letter block $a_1a_2 \dots a_s$, then we *would like* to label the unitary message $a_1a_2 \dots a_s$ by the string of integers $x_1x_2 \dots x_s$, where x_i , with $1 \leq i \leq s$, is the numerical equivalent of a_i . Why are we prevented from doing so? Unfortunately, there is a notational ambiguity which we want to draw attention to.

Indeed, assume we want to transmit a message consisting of 2-letter blocks, using the numerical equivalent of the letters. Then the numerical sequence 114 might correspond to the message *bo*, but also to the message *le*, depending on whether we look at the number 114 as consisting of 1 and 14 or 11 and 4. This is due to having a correspondence with numbers not all consisting of the same number of digits.

When the unitary message consists of more than one letter, in order to avoid ambiguities we may use other correspondences: for instance 2-digit numerical equivalents for the letters or binary numerical equivalents, as described by Table 7.5.

Table 7.5. 2-digit and binary numerical equivalents

a \rightarrow 00 = 0000	j \rightarrow 09 = 01001	s \rightarrow 18 = 10010
b \rightarrow 01 = 00001	k \rightarrow 10 = 01010	t \rightarrow 19 = 10011
c \rightarrow 02 = 00010	l \rightarrow 11 = 01011	u \rightarrow 20 = 10100
d \rightarrow 03 = 00011	m \rightarrow 12 = 01100	v \rightarrow 21 = 10101
e \rightarrow 04 = 00100	n \rightarrow 13 = 01101	w \rightarrow 22 = 10110
f \rightarrow 05 = 00101	o \rightarrow 14 = 01110	x \rightarrow 23 = 10111
g \rightarrow 06 = 00110	p \rightarrow 15 = 01111	y \rightarrow 24 = 11000
h \rightarrow 07 = 00111	q \rightarrow 16 = 10000	z \rightarrow 25 = 11001
i \rightarrow 08 = 01000	r \rightarrow 17 = 10001	

Table 7.6. ASCII code

32	44 ,	56 8	68 D	80 P	92 \	104 h	116 t
33 !	45 -	57 9	69 E	81 Q	93]	105 i	117 u
34 "	46 .	58 :	70 F	82 R	94 ^	106 j	118 v
35 #	47 /	59 ;	71 G	83 S	95 _	107 k	119 w
36 \$	48 0	60 <	72 H	84 T	96 ‘	108 l	120 x
37 %	49 1	61 =	73 I	85 U	97 a	109 m	121 y
38 &	50 2	62 >	74 J	86 V	98 b	110 n	122 z
39 ’	51 3	63 ?	75 K	87 W	99 c	111 o	123 {
40 (52 4	64 @	76 L	88 X	100 d	112 p	124
41)	53 5	65 A	77 M	89 Y	101 e	113 q	125 }
42 *	54 6	66 B	78 N	90 Z	102 f	114 r	126 ~
43 +	55 7	67 C	79 O	91 [103 g	115 s	

By using the binary or 2-digit numerical correspondence, the ambiguity disappears. For instance, the messages *le* and *bo*, which had the same numerical equivalent, now have different ones:

	Num. eq.	2-digit	Binary
le	114	1104	0101100100
bo	114	0114	0000101110

In next section we shall discuss further how to avoid this ambiguity. However, notice that for the sake of simplicity we might keep using the standard numerical correspondence using the simple device of separating with spaces the numbers corresponding to different letters. This is the method we shall use in the simplest examples. For instance, the message *CIAO* will be transcribed 2 8 0 14 rather than 28014.

Another standard way of associating with each alphabet letter and with each character a number is described, as in the *American Standard Code for Information Interchange* (ASCII), by Table 7.6: it is a usual code used to translate the symbols, more commonly employed when inputting a text into a computer. In Table 7.6 numbers start from 32, as the integers smaller than 32 represent special control characters affecting the operation of the computer.

7.4 Some classic ciphers based on modular arithmetic

We shall now describe the mathematical aspects of some of the ciphers seen up to now. For each of them we shall give some examples according to the following pattern:

- fix the length of the unitary message, appending at the end of the message, if necessary, the letter *x* a number of times sufficient for the whole message to have a length suitable to divide it in blocks of the same length;

- transform the blocks into numerical equivalents following a procedure to be described;
- choose a key k and a corresponding cipher: that is, define the function C_k that determines the cipher on the alphabet in its numerical form;
- determine $D_k = C_k^{-1}$;
- reconstruct the message in the usual alphabet.

Remark 7.4.1. Let N be the length of the alphabet. As remarked, usually we take $N = 26$. Once the length s of the unitary message consisting of s letters is fixed, if we call x_1, \dots, x_s the numerical equivalents of these letters, we describe a procedure that uniquely determines what shall be written taking x_1, \dots, x_s as starting point. We shall associate with the block $x_1 \dots x_s$ the number a that in base N is $(x_1 \dots x_s)_N$. We know that $a \in \{0, \dots, N^s - 1\}$, that is to say, we may identify the set of unitary messages (s -letter blocks) with \mathbb{Z}_{N^s} . So it is clear that an enciphering is just an invertible function on, and taking values in, \mathbb{Z}_{N^s} . In the previous example, where $N = 26$, we shall have

$$le \longrightarrow 26 \cdot 11 + 4 = 290, \quad bo \longrightarrow 26 \cdot 1 + 14 = 40.$$

On the other hand, if we begin with the number 40, knowing that $N = 26$, we find that the corresponding plaintext is bo and only this. So we have solved in yet another way the ambiguity issue mentioned in the previous section.

An alternative way of denoting the s -letter block $x_1 \dots x_s$ with no ambiguity is to represent it as an element of $\mathbb{Z}_N^s = \underbrace{\mathbb{Z}_N \times \mathbb{Z}_N \times \dots \times \mathbb{Z}_N}_s$, as each x_i is in \mathbb{Z}_N .

This defines an obvious bijection between \mathbb{Z}_N^s and \mathbb{Z}_{N^s} given by

$$\begin{aligned} \mathbb{Z}_N^s &= \underbrace{\mathbb{Z}_N \times \mathbb{Z}_N \times \dots \times \mathbb{Z}_N}_s \rightarrow \mathbb{Z}_{N^s}, \\ (x_1, x_2, \dots, x_s) &\mapsto x_1 + x_2 N + \dots + x_{s-1} N^{s-2} + x_s N^{s-1} = (x_s \dots x_1)_N. \end{aligned} \quad (7.1)$$

So, in general, if we split up the message in s -letter blocks, on an N -letter alphabet, the enciphering function is a bijection

$$f : \mathbb{Z}_N^s \longrightarrow \mathbb{Z}_{N^s}.$$

A feature of the cipher we are going to describe in this section, the “classic” ciphers, is that the deciphering key D_k is easily computed from the enciphering key C_k . In other words, from a computational viewpoint, the knowledge of the deciphering key is *essentially equivalent* to the knowledge of the enciphering key. In public key ciphers we shall describe later, which rely on very different mathematical ideas, it is possible, on the contrary, to divulge the enciphering key without compromising the secrecy of D_k . Indeed, in these systems, computing D_k from C_k is so computationally hard to be unfeasible in practice. For these reasons, classic ciphers are called *two-way* or *symmetric*,

while public key ciphers are also called *one-way* or *asymmetric*. But more on this later.

Let us now examine systematically some kinds of classic ciphers, all relying on modular arithmetic, which admit as particular instances those considered above. For the sake of simplicity, in the examples we shall use the numerical equivalents, inserting spaces between the numbers to avoid any ambiguity.

7.4.1 Affine ciphers

Assume the unitary message consists of a single letter, that is to say, the numerical alphabet of the messages is $\mathcal{P} = \mathbb{Z}_{26}$. If P is a letter, we shall also denote by P its numerical equivalent. We shall use the same convention for the letters C in the ciphertext.

Affine ciphers are described by an enciphering function that uses an *affine transformation*, that is a bijection

$$C_k : \mathbb{Z}_{26} \longrightarrow \mathbb{Z}_{26}, \quad P \longrightarrow (aP + b) \bmod 26,$$

where $a, b \in \mathbb{Z}$ and the pair $k = (a, b)$ represents the *key* of the system. For C_k to be bijective, it is necessary for a to be relatively prime with 26, that is, $\text{GCD}(a, 26) = 1$ (see Exercise A7.3). In this case the congruence $xa \equiv 1 \pmod{26}$ has a unique solution a' modulo 26. Then the inverse deciphering function D_k is

$$D_k = C_k^{-1} : \mathbb{Z}_{26} \longrightarrow \mathbb{Z}_{26}, \quad C \longrightarrow a'(C - b) \bmod 26.$$

Consider now a key $k = (a, b) \in \mathcal{K}$. Notice that *Caesar* or *translation ciphers* are affine ciphers with $a = 1$.

Clearly, we may assume $0 \leq b \leq 25$ and $1 \leq a \leq 25$, taking a and b modulo 26. Recalling that a is relatively prime with 26 if and only if a is an invertible element in \mathbb{Z}_{26} , that is, if $a \in U(\mathbb{Z}_{26})$, it follows that the key set is

$$\mathcal{K} = U(\mathbb{Z}_{26}) \times \mathbb{Z}_{26}.$$

How many affine ciphers are there? In other words, how many elements are there in \mathcal{K} ?

Recall (see § 3.3 and § 4.2.1) that

$$|U(\mathbb{Z}_{26})| = \varphi(26) = |\{1 \leq a \leq 25 \mid \text{GCD}(a, 26) = 1\}| = 12;$$

so there are $12 \cdot 26 = 312$ affine ciphers, including a trivial one, corresponding to $k = (1, 0)$.

In Table 7.7 on page 337 we give a step-by-step example of one of these ciphers, with $a = 7$ and $b = 10$, that is, $k = (7, 10)$. Notice that a Bézout's identity for 7 and 26 is

$$-11 \cdot 7 + 3 \cdot 26 = 1,$$

as can be found by applying the Euclidean algorithm (see § 1.3.3), and so $a' = -11 \equiv 15 \pmod{26}$ and the deciphering function is

$$D_{(7,10)}(C) = 15(C - 10) \bmod 26,$$

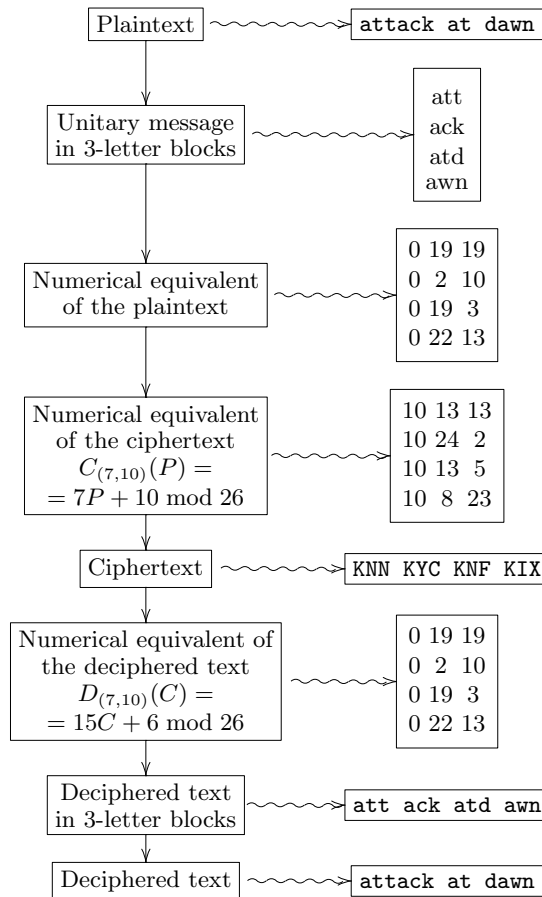
that is, $D_{(7,10)} = C_{(15,6)}$, as $-150 \equiv 6 \pmod{26}$.

Suppose we have intercepted a message which is known to be in English and to have been enciphered with this system. How do we decrypt it? That is, how do we find the coefficients a and b of the affine transformation? Once more, with a frequency analysis.

Assume that in the ciphertext the two letters appearing with the least frequency are R and S : it stands to reason to guess that these letters correspond, in the plaintext, to j or z . If we suppose that $R \leftrightarrow 17$ corresponds to the letter $q \leftrightarrow 16$, and $S \leftrightarrow 18$ corresponds to $z \leftrightarrow 25$, then, according to the relation $C = aP + b$, the following congruence system must hold

$$\begin{cases} 17 \equiv a \cdot 16 + b & (\text{mod } 26), \\ 18 \equiv a \cdot 25 + b & (\text{mod } 26). \end{cases}$$

Table 7.7. Affine enciphering with $k = (7, 10)$ modulo 26



An effective way of describing the system is using matrices. In this way, we may write down the system as

$$A \cdot \begin{pmatrix} a \\ b \end{pmatrix} \equiv \begin{pmatrix} 17 \\ 18 \end{pmatrix} \pmod{26}, \quad \text{with } A = \begin{pmatrix} 16 & 1 \\ 25 & 1 \end{pmatrix}.$$

To solve the system it is necessary for the matrix A to be invertible modulo 26. In fact, it is easy to prove the following proposition (see Exercise A7.7).

Proposition 7.4.2. *Let*

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1s} \\ a_{21} & a_{22} & \dots & a_{2s} \\ \vdots & \vdots & \ddots & \vdots \\ a_{s1} & a_{s2} & \dots & a_{ss} \end{pmatrix}$$

with $a_{ij} \in \mathbb{Z}_N$. The following are equivalent:

- $\text{GCD}(\det(A), N) = 1$;
- A is invertible, that is, there is a unique matrix A^{-1} defined over \mathbb{Z}_N such that $A \cdot A^{-1} = A^{-1} \cdot A$ is the identity matrix. The matrix A^{-1} is given by

$$A^{-1} = \det(A)^{-1} \begin{pmatrix} A_{11} & A_{21} & \dots & A_{s1} \\ A_{12} & A_{22} & \dots & A_{s2} \\ \vdots & \vdots & \ddots & \vdots \\ A_{1s} & A_{2s} & \dots & A_{ss} \end{pmatrix}$$

where $\det(A)^{-1}$ is the inverse of $\det(A)$ in \mathbb{Z}_N and A_{ij} denotes the cofactor corresponding to the element a_{ij} in A ;

- the map $f : X \in \mathbb{Z}_N^s \rightarrow A \cdot X \in \mathbb{Z}_N^s$ is bijective ($X \in \mathbb{Z}_N^s$ is thought of as a column vector of order s);
- for every column vector $Y \in \mathbb{Z}_N^s$, the system $A \cdot X = Y$ has a unique solution.

Notice that if N is prime, that is, if we are working in a field, the condition for A to be invertible is $\det(A) \neq 0$. In this case, Proposition 7.4.2 is a well-known result in linear algebra.

Coming back to our example, in which $\det A = -9 \equiv 17 \pmod{26}$ and $\text{GCD}(17, 26) = 1$, we have

$$A^{-1} \equiv \begin{pmatrix} 23 & 3 \\ 23 & 4 \end{pmatrix} \pmod{26}$$

is the inverse modulo 26 of the system matrix. So the solutions a and b are immediately found as follows:

$$\begin{pmatrix} a \\ b \end{pmatrix} \equiv A^{-1} \cdot \begin{pmatrix} 17 \\ 18 \end{pmatrix} \equiv \begin{pmatrix} 3 \\ 21 \end{pmatrix} \pmod{26}.$$

In this particular case the simplest way of solving the original system would have been to subtract the second equation from the first one, immediately finding $a \equiv 3 \pmod{26}$ and then $b \equiv 21 \pmod{26}$. But we have given the general solution method for this kind of problems.

Remark 7.4.3. A word of caution is necessary about the cryptanalytical method to find a and b just described. It leads to linear congruence systems of the form

$$A \cdot \begin{pmatrix} a \\ b \end{pmatrix} \equiv \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \pmod{m},$$

where α, β and the square 2×2 matrix A are known. If the determinant of A is invertible modulo m , then A has an inverse modulo m and the system admits a unique solution (a, b) , given by

$$\begin{pmatrix} a \\ b \end{pmatrix} \equiv A^{-1} \cdot \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \pmod{m}.$$

The reader is encouraged to verify this claim and to extend it to systems in several unknowns (see Proposition 7.4.2 and Exercise A7.8).

If, on the other hand, the determinant of A is not invertible modulo m , the system might have no solutions (see Exercise A7.4) or more than one solution (see Exercise A7.5). In the first case this means that our guesses about frequency analysis are certainly wrong and we shall make different cryptanalytical attempts.

If there is more than one solution, we might try out each one of them and check whether it works. For instance, if the determinant of A is not invertible modulo m but is invertible modulo a prime number q divisor of m , we might solve the problem modulo q . This shall give a unique solution modulo q , but several solutions modulo m , each one to be analysed to check whether it works (see Exercise A7.6).

We conclude with an example, in the context of Caesar ciphers, which uses an enciphering in \mathbb{Z}_N^s , leaving as an exercise an example that relies on the identification of \mathbb{Z}_N^s with \mathbb{Z}_{N^s} described by (7.1).

Example 7.4.4. Assume the numerical alphabet of the unitary messages to be represented by $\mathcal{P} = \mathbb{Z}_{26}^s$ with s a fixed integer, that is, the unitary messages to consist of an s -letter block. Given the key $k \in \mathcal{K} = \{1, 2, \dots, 25\}$, as we have seen, the enciphering function is

$$C_k : \mathbb{Z}_{26}^s \longrightarrow \mathbb{Z}_{26}^s, \quad p \longmapsto p + \mathbf{k} \pmod{26},$$

where p is the unitary message having numerical entries $p_1 \dots p_s$, and $\mathbf{k} = (k, \dots, k)$ is the element of \mathbb{Z}_{26}^s having all entries equal to k . By mod 26, it is meant that each entry of an element of \mathbb{Z}_{26}^s is computed modulo 26. Notice that if we take as our key a vector $(k_1, \dots, k_s) \in \mathbb{Z}_{26}^s$, where k_1, \dots, k_s are not all equal, then we construct a polyalphabetic cipher; we shall deal with it again later.

Table 7.8 on page 340 shows an example with $k = 5$.

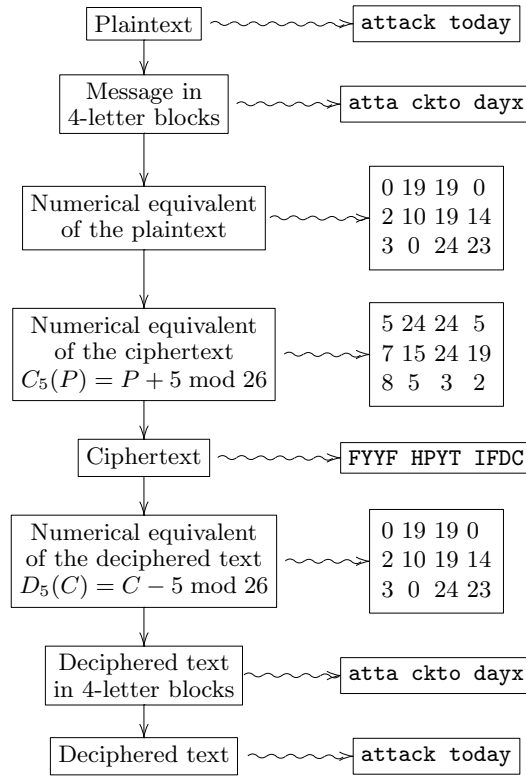
If, instead, we identify an element (p_1, \dots, p_s) of \mathbb{Z}_{26}^s with the number written in base 26 as

$$p = p_1 + p_2 \cdot 26 + p_3 \cdot 26^2 + \dots + p_s \cdot 26^{s-1},$$

where, clearly, p_1, \dots, p_s are in $\{0, 1, \dots, 25\}$, we may, as remarked above, identify \mathbb{Z}_{26}^s with \mathbb{Z}_{26^s} via the map

$$(p_1, \dots, p_s) \in \mathbb{Z}_{26}^s \longrightarrow p = p_1 + p_2 \cdot 26 + \dots + p_s \cdot 26^{s-1} \in \mathbb{Z}_{26^s}.$$

So we have $\mathcal{P} = \mathbb{Z}_{26^s}$ and $\mathcal{K} = \mathbb{Z}_{26^s}$. Exercise B7.25 uses this different enciphering, with key $k = 100$.

Table 7.8. Translation enciphering with $k = 5$ 

7.4.2 Matrix or Hill ciphers

These ciphers split up the text into blocks of length s , translate each letter of the block into its numerical equivalent and then apply an enciphering function, defined on the blocks, of the form

$$c = Ap + b \bmod 26, \quad (7.2)$$

where A is a square $s \times s$ matrix, b is a fixed column vector of length s , and p and c are the column vectors corresponding numerically to plaintext p and ciphertext c . Moreover, if we want to be able to decipher the message, that is, have a bijective enciphering function, it is necessary for the matrix A to be invertible modulo 26. The map

$$C_{(A,b)} : \mathbb{Z}_{26}^s \rightarrow \mathbb{Z}_{26}^s, \quad p \rightarrow Ap + b \bmod 26$$

is also called an *affine transformation* defined by the *key* $k = (A, b)$.

Notice that in the case $s = 1$ we find again the affine ciphers described above.

Example 7.4.5. We conclude giving an example of affine transformation defined by the key

$$k = (A, b), \quad \text{with } A = \begin{pmatrix} 1 & 2 \\ 4 & 3 \end{pmatrix}, \quad b = 0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

So the enciphering function is

$$C_{(A,0)} \begin{pmatrix} p_1 \\ p_2 \end{pmatrix} = \begin{pmatrix} 1 & 2 \\ 4 & 3 \end{pmatrix} \cdot \begin{pmatrix} p_1 \\ p_2 \end{pmatrix} = \begin{pmatrix} p_1 + 2p_2 \\ 4p_1 + 3p_2 \end{pmatrix} \pmod{26}.$$

Notice that A is invertible modulo 26, as $\det(A) = -5 \equiv 21 \pmod{26}$ is relatively prime with 26. So we may compute the inverse modulo 26 of A (see Proposition 7.4.2). The inverse of 21 modulo 26 is 5, as Bézout's identity found with the Euclidean algorithm is

$$5 \cdot 21 - 4 \cdot 26 = 1.$$

So the inverse modulo 26 of A is

$$A^{-1} = 5 \cdot \begin{pmatrix} 3 & -2 \\ -4 & 1 \end{pmatrix} = \begin{pmatrix} 15 & -10 \\ -20 & 5 \end{pmatrix} \equiv \begin{pmatrix} 15 & 16 \\ 6 & 5 \end{pmatrix} \pmod{26}$$

and the deciphering function is

$$D_{(A,0)} \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} 15 & 16 \\ 6 & 5 \end{pmatrix} \cdot \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} 15c_1 + 16c_2 \\ 6c_1 + 5c_2 \end{pmatrix} \pmod{26}.$$

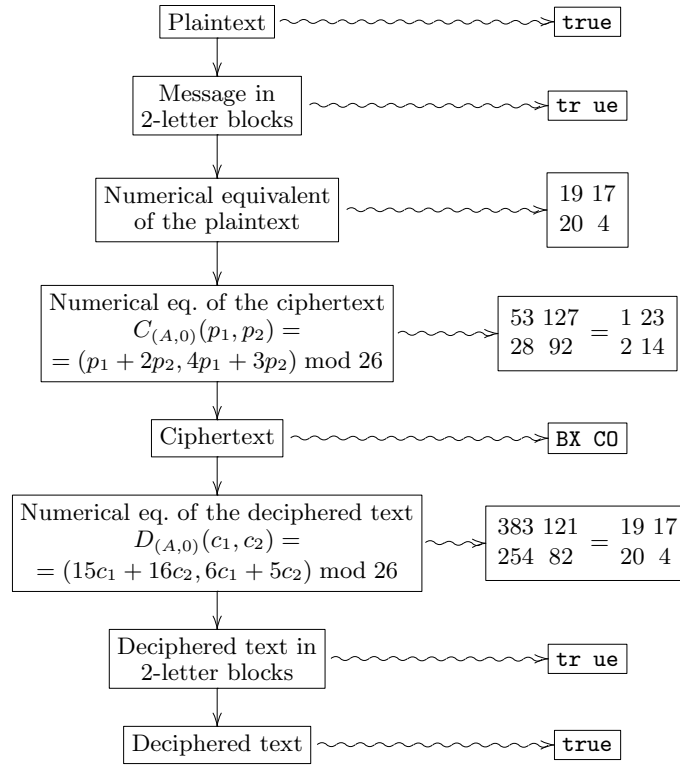
In Table 7.9 we show the complete procedure to encipher with the key given the plaintext **true**.

7.5 The basic idea of public key cryptography

In this section we are going to continue the description of some cryptographic systems, outlining the genesis of the so-called public key systems. Later we shall illustrate specific cryptographic systems of this kind, like the system based on the knapsack problem and the *RSA* system: the security of the former relies on the difficulty of some combinatorial problems, while that of the latter on the difficulty of factoring large numbers.

In the ciphers described so far the deciphering procedure is not difficult, once the enciphering method, and so the key, are known. In fact, in those cases the deciphering function is, in a way, *symmetric* with respect to the enciphering function: it is, both computationally and logically, a function of the same kind. In particular, all classic cryptosystems concern the exchange of messages between *two* users and rely on exchanging a *key* which, basically, enables both enciphering and deciphering.

In an age like the present one, when most information is transmitted by telephone or electronic mail or radio, every sent message, as well as every sent key, is susceptible to being easily eavesdropped. Moreover, it is necessary to make it possible to communicate for users who have never met and so have not had, in principle, the opportunity of exchanging private enciphering keys. So it is indispensable to find new, and more secure, ways of enciphering messages. This is the goal of public key cryptography.

Table 7.9. Matrix cipher as in Example 7.4.5

A public key cipher is a cipher that allows both the method employed and the enciphering key to be made public - hence the name of *public key cipher* - without revealing how to decipher the messages. In other words, in these systems, to be able to compute in a *reasonably short* time the deciphering transformation, which is the inverse of the enciphering one, it is necessary to be in possession of a further piece of information, besides the public ones. So this information is kept secret and without it the complexity of the deciphering is enough to make it unfeasible: in essence, to decipher without further information would require a time exceedingly long with respect to the time required to encipher.

Remark 7.5.1. For an example which illustrates quite well the fact that being able to do something does not imply being able to perform the inverse operation, consider the telephone directory of a big city. It is easy to look up the telephone number of a certain person, but it might be impossible, that is to say, it might take too long with respect to the available time, to trace a person from his number.

From a mathematical viewpoint, carrying out this idea relies on the notion of *one-way function*.

We shall call a function $f : S \rightarrow T$ from a set S to a set T *one-way* if it can be computed easily (for instance, because it is computed in polynomial time), but, having chosen a random $y \in f(S)$, it is computationally much harder, and impossible in practice (for instance because it takes an exponential time), to find an $x \in S$ such that $y = f(x)$.

This notion may appear quite vague, as it uses terms as “easy”, “chosen a random $y \in f(S)$ ”, or “impossible in practice”, which have not the mathematical rigour of a definition. Nevertheless, we believe that it gives a sufficiently clear idea of the meaning of a one-way function.

Example 7.5.2. Consider a finite group G of order n and an element $b \in G$. Set

$$S = \mathbb{Z}_n = \{0, 1, \dots, n-1\};$$

then we may consider the *exponential function*

$$f : S \rightarrow G, \quad f(x) = b^x.$$

If $y = f(x)$, we call x a *discrete logarithm* of y over G in base b and denote it by the symbol $\log_b y$. When G is the multiplicative group \mathbb{F}_q^* of a finite field \mathbb{F}_q , if $b \in \mathbb{F}_q^*$ is one of its generators, then f is bijective and its inverse function is called *discrete logarithm* over \mathbb{F}_q in base b . In this case, computing f requires a polynomial time (see Proposition 5.1.44). On the other hand, all known algorithms to compute discrete logarithms are exponential and it is conjectured that there are no polynomial ones. So the exponential function over \mathbb{F}_q can be regarded as a one-way function. However, it must be remarked that some algorithms to compute discrete logarithms, one example of which we shall shortly illustrate with the so-called *Baby step–giant step* algorithm, are, in particular cases, quite effective.

In general, in cryptography it is interesting to consider those groups G for which, as for \mathbb{F}_q^* , computing powers is computationally easy (for instance, requiring polynomial time), while computing discrete logarithms is computationally far harder (for instance, exponential). This may yield one-way functions.

Example 7.5.3. Let \mathbb{F}_q be a *large enough* finite field, that is of order $q = p^f$ with p a large prime number, and let $f(x) \in \mathbb{F}_q[x]$ be a polynomial such that the corresponding polynomial function $f : \mathbb{F}_q \rightarrow \mathbb{F}_q$ is injective. As we are well aware, the function f can be computed in polynomial time, while the inverse function f^{-1} may be quite hard to compute in practice. It is conjectured that such a polynomial function is in many cases a one-way function.

The next example describes another kind of function f which is computed more easily than f^{-1} , but in which, unlike exponential and logarithmic functions, both are computed in polynomial time.

Example 7.5.4. We may consider enciphering a message $X \in \mathbb{Z}_N^s$ by multiplying it on the left by a square matrix A of order s . We have

$$f : X \in \mathbb{Z}_N^s \rightarrow A \cdot X \in \mathbb{Z}_N^s.$$

Computing $Y = f(X) = A \cdot X$ requires about s^2 operations (see Exercise A2.19). Computing $X = f^{-1}(Y) = A^{-1}Y$, on the contrary, has a much greater computational cost, as it requires inverting a square matrix of order s , which implies about s^3 operations (see Exercise A2.22).

The existence of one-way functions has not yet been rigorously proved. However, there are many good candidates, like the exponential functions on finite fields, mentioned above. In practice, we are interested in a specific kind of one-way functions that can be defined in a vague but sufficiently eloquent way, as follows:

Definition 7.5.5. A one-way function $f : S \rightarrow T$ is said to be a trapdoor function if with some further information it becomes computationally feasible to find, for all $y \in f(S)$, an element $x \in S$ such that $f(x) = y$.

Public key techniques make use of functions of this kind, in the sense that, basically, they are used as enciphering functions. We shall shortly show two examples which shall illustrate this basic idea, which has remained so far quite indeterminate.

7.5.1 An algorithm to compute discrete logarithms

We devote a short section to describe an algorithm, the so-called *Baby step-giant step algorithm*, to compute discrete logarithms over the field \mathbb{Z}_p with p a prime number. So we work in \mathbb{Z}_p^* , whose elements shall be identified with $1, 2, \dots, p-1$. Let g be a generator of \mathbb{Z}_p^* . We want to determine the discrete logarithm x of $y \in \mathbb{Z}_p$ in base g . We proceed as follows:

- **baby steps:** set n equal to the least integer greater than \sqrt{p} and compute the values $g^i \in \{1, \dots, p-1\}$, for all $i \in \{0, \dots, n-1\}$, inserting them in a list to be kept in the memory;
- **giant steps:** compute g^n and then g^{-n} and successively $yg^{-n}, yg^{-2n}, yg^{-3n}, \dots, yg^{-n^2}$. After each of these computations, compare the result with the numbers in the list created in the first step. As soon as we obtain an equality of the form $yg^{-jn} = g^i$, we have found the logarithm $x = jn + i$.

First of all, notice that the algorithm terminates and gives the desired logarithm. In fact, the logarithm exists and is a number x in $\{1, \dots, p-1\}$. Then, dividing x by n we have $x = nj + i$, with $0 \leq i \leq n-1$. On the other hand, as $x < p < n^2$ we also have $1 \leq j \leq n$.

We estimate next the complexity of the algorithm. There are n *baby steps*, each having complexity $\mathcal{O}(\log n)$. So the total complexity of the *baby steps* is $\mathcal{O}(n \log n)$. Notice that the *baby steps* can be thought of as a kind of *precomputation*, in the

sense that they are performed just once, independently of the number y of which we are computing the logarithm.

Reasoning in a similar way, we see that the *giant steps* too, which do depend on the number whose logarithm we are computing, have complexity $\mathcal{O}(n \log n)$; so this is the complexity of the whole algorithm. It is exponential in n . There are further issues with this algorithm:

- the algorithm needs a large amount of memory if p is large, as a list consisting of $\lfloor \sqrt{p} \rfloor + 1$ integers must be kept in memory;
- moreover, comparing the numbers in this list and the numbers computed in the *giant steps* has a computational cost, even if we have neglected it so far. A possible way to perform this comparison consists in dividing a number by the other one and checking whether the quotient is greater than zero or not. Clearly, the algorithm leads us to carry out n^2 comparisons, so all of them taken together yield an exponential complexity.

Example 7.5.6. Let us illustrate the above by means of a very simple example. Take $p = 11$ and $g = 2$, which is a generator of \mathbb{Z}_{11}^* . In this case we have $3 < \sqrt{11} < 4$, so $n = 4$. The baby steps yield the list

$$2^0 = 1, \quad 2^1 = 2, \quad 2^2 = 4, \quad 2^3 = 8. \quad (7.3)$$

Compute next 2^4 , which equals 5 modulo 11, while its inverse is 9.

Suppose we want to compute the logarithm $\log_2 6$. So we perform the giant steps. In the first step we compute $6 \cdot 2^{-4}$, which equals 10 modulo 11. As this number is not included in the list (7.3), we have not found the required logarithm. Perform another giant step, computing $6 \cdot 2^{-8}$, which, modulo 11, equals 2. So we find the relation $6 \cdot 2^{-8} = 2$, or $6 = 2^9 \pmod{11}$, that is, $\log_2 6 = 9$.

For other algorithms to compute discrete logarithms, see [30], Ch. IV, or [44].

7.6 The knapsack problem and its applications to cryptography

Suppose we are about to leave for an excursion. We have to pack our knapsack and we want to make maximum use of the available space. We have a number, say n , of different objects, having volume v_1, v_2, \dots, v_n ; we know that the knapsack contains a volume V , and we want to carry the greatest possible load. How do we find it? We are looking for a subset $J \subseteq \{1, 2, \dots, n\}$ such that

$$V = \sum_{j \in J} v_j. \quad (7.4)$$

This scheme may be applied to several similar problems. Assume we have to pay 2 Euros and have at our disposal 2-, 5-, and 10-cent coins: how can we pay with the least number of coins? Or with the largest number, so as to rid ourselves of as many coins as possible? Moreover, how many possible ways are there of paying?

Let us turn back to the knapsack problem and describe a cryptographic system relying on it, devised by Merkle and Hellman in 1978.

First of all, rephrase the problem as follows. Given n positive integers a_1, a_2, \dots, a_n and a positive integer m , can we find n integers x_1, x_2, \dots, x_n with $x_i \in \{0, 1\}$ so that our integer m can be written as

$$m = a_1x_1 + a_2x_2 + \dots + a_nx_n? \quad (7.5)$$

In other words, is it possible to write m as a sum of *some* of the a_i s? It is not always possible and a solution, if it exists, may or may not be unique. The three following examples demonstrate the different possible cases.

Example 7.6.1. Set $n = 5$, $(a_1, a_2, \dots, a_5) = (2, 7, 8, 11, 12)$ and $m = 21$. It is immediate to see that $21 = 2 + 8 + 11$ and $21 = 2 + 7 + 12$; so we have two solutions, and they are the only ones. In explicit form, the first solution is $x_1 = x_3 = x_4 = 1$ and $x_2 = x_5 = 0$, while the second one is $x_1 = x_2 = x_5 = 1$ and $x_3 = x_4 = 0$.

Example 7.6.2. Consider now the same a_i s as before, but $m = 1$. As m is smaller than each of the a_i s, it is not possible to write m as a combination of the a_i s with coefficients 0 or 1.

Example 7.6.3. If $a_i = 2^{i-1}$, for $i = 1, \dots, n$, solving the knapsack problem means finding *the binary representation of m* which, as we know, exists and is unique.

In principle, in order to find the solution, if it exists, it suffices to consider all the sums of the form (7.4) with J ranging among all the subsets of $\{1, \dots, n\}$. As is well known (see Exercise A1.22), there are 2^n such subsets, including the empty set.

If n is small, this kind of inspection can be carried out, but if n is large, it is computationally unfeasible, as it is likely to require an exponential algorithm. In general, in fact, no algorithm to solve the knapsack problem is known, apart from trying out all the possibilities.

We may ask if there are *integer* solutions $x_i \in \mathbb{N}$ of Equation (7.5). However, this general formulation of the knapsack problem is beyond the scope of this text.

Remark 7.6.4. The knapsack problem is known to belong to a category of very hard problems, the so-called \mathcal{NP} -problems, for which it is conjectured that *no algorithm giving the solution in polynomial time exists*.

More in detail, let \mathcal{P} be the class of problems P for which a deterministic algorithm that solves P in polynomial time exists. We have seen so far several examples of problems lying in class \mathcal{P} : for instance, the problem of finding the greatest common divisor of two integer numbers, or that of recognising whether a number is prime.

A problem P is said to belong to class \mathcal{NP} if there are algorithms - not necessarily polynomial ones - solving it and if it is possible to verify whether given data solve the problem or not, using a polynomial deterministic algorithm.

For instance, the problem of factoring an integer number n is of this kind. The sieve of Eratosthenes is a non-polynomial algorithm solving the problem, while, given a number m , we can verify in polynomial time, using the Euclidean algorithm, whether m divides n or not.

It is easy to see that the knapsack problem is too in \mathcal{NP} .

Clearly, $\mathcal{P} \subseteq \mathcal{NP}$; the main conjecture in complexity theory states that $\mathcal{P} \neq \mathcal{NP}$.

A problem P in \mathcal{NP} is said to be \mathcal{NP} -complete if, for every other problem Q in \mathcal{NP} , there is a polynomial deterministic algorithm that reduces solving Q to solving P . Clearly, if P is \mathcal{NP} -complete and if there were a polynomial deterministic algorithm that solves P , then every problem in \mathcal{NP} would also be in \mathcal{P} . So, if the main conjecture in complexity theory is true, there are no polynomial deterministic algorithms that solve \mathcal{NP} -complete problems. These problems are, basically, the *most computationally difficult* problems in \mathcal{NP} .

As we said at the beginning of this remark, the knapsack problem is known to be \mathcal{NP} -complete (see [23]).

A special case of our problem is the one in which the sequence a_1, a_2, \dots, a_n is *superincreasing*, in the sense of the following definition.

Definition 7.6.5. A sequence of n positive integers a_1, a_2, \dots, a_n is superincreasing if the following inequalities hold

$$\begin{aligned} a_1 &< a_2, \\ a_1 + a_2 &< a_3, \\ a_1 + a_2 + a_3 &< a_4, \\ &\vdots \\ a_1 + a_2 + \dots + a_{n-1} &< a_n. \end{aligned}$$

Is there a solution to the knapsack problem in this case? The answer is not always in the affirmative but, if a solution exists, then it is unique and can be found in polynomial time. Indeed, to find the value x_1, \dots, x_n such that $m = \sum_{i=1}^n x_i a_i$ with $x_i \in \{0, 1\}$ the following algorithm may be used

As a first thing, determine x_n , by noticing that necessarily:

$$x_n = \begin{cases} 1 & \text{if } m \geq a_n, \\ 0 & \text{if } m < a_n. \end{cases}$$

To determine x_{n-1} , do the same, substituting $m - x_n a_n$ for m . In other words, we look for a solution to the knapsack problem by trying to express $m - x_n a_n$ as

$$m - a_n x_n = a_1 x_1 + a_2 x_2 + \dots + a_{n-1} x_{n-1}.$$

Notice that, clearly, the sequence a_1, a_2, \dots, a_{n-1} is superincreasing too. So we have

$$x_{n-1} = \begin{cases} 1 & \text{if } m - x_n a_n \geq a_{n-1}, \\ 0 & \text{if } m - x_n a_n < a_{n-1}. \end{cases}$$

In general, having found x_n, \dots, x_{j+1} , we shall set

$$x_j = \begin{cases} 1 & \text{if } m - \sum_{i=j+1}^n x_i a_i \geq a_j, \\ 0 & \text{if } m - \sum_{i=j+1}^n x_i a_i < a_j. \end{cases}$$

It is clear that if $m - \sum_{i=j+1}^n x_i a_i = 0$, we have found the solution, which is clearly unique. If on the other hand $m - \sum_{i=j+1}^n x_i a_i > 0$ but a_j, \dots, a_1 are greater than $m - \sum_{i=j+1}^n x_i a_i$, then no solution exists.

It is not hard to see that the algorithm just described is of polynomial type (see Exercise A7.9).

Let us illustrate with an example this algorithm.

Example 7.6.6. Consider the superincreasing sequence $(1, 4, 6, 13, 25)$. How do we compute the solution when $m = 26$ according to the algorithm?

As the rightmost term of our sequence is 25 and $25 < 26$, then we have to choose $x_5 = 1$. Now we carry on the procedure with $26 - 1 \cdot 25 = 1 < 13$, so we set $x_4 = 0$. As $26 - 1 \cdot 25 - 0 \cdot 13 = 1 < 6$, then $x_3 = 0$, hence again $x_2 = 0$. The last step, applied to $26 - 1 \cdot 25 - 0 \cdot 13 - 0 \cdot 6 - 0 \cdot 4 = 1 = a_1$ gives $x_1 = 1$ and so the solution. In fact, we have

$$26 = 1 \cdot 1 + 0 \cdot 4 + 0 \cdot 6 + 0 \cdot 13 + 1 \cdot 25.$$

Notice that if the sequence were $(2, 3, 6, 13, 25)$, there would have been no solution, as in the last step we should have written 1 as $x_1 \cdot 2$, which is not possible. In other words, the solution does not exist because $a_1 = 2 > 26 - 1 \cdot 25 - 0 \cdot 13 - 0 \cdot 6 - 0 \cdot 3$.

Let us see now how to construct a *cipher* related to the knapsack problem.

7.6.1 Public key cipher based on the knapsack problem, or Merkle–Hellman cipher

The cipher consists of the following steps.

- Each user X chooses a superincreasing sequence a_1, a_2, \dots, a_N of a fixed length N , an integer m such that $m > 2a_N$, and an integer w relatively prime with m . These data are kept *secret*.
- User X computes the transformed sequence

$$b_j = wa_j \bmod m, \quad \text{for } j = 1, \dots, N.$$

The sequence b_1, b_2, \dots, b_N is made *public* by X and is the *enciphering key*.

- A user Y may send a message p to X acting as follows. First of all, he transforms each letter of the text into its binary equivalent using Table 7.5 on page 333. Next, he splits up the resulting sequence of 0s and 1s into blocks of length N , adding at the end, if necessary, a number of 1s so as to have blocks all of the same length N . For each block, say $p = x_1x_2 \dots x_N$, the user Y applies the transformation to encipher $p \rightarrow c = b_1x_1 + \dots + b_Nx_N$, and sends the enciphered text c to X .
- How has X to proceed to decipher Y 's message? As a first thing, X computes the *deciphering key*, that is, $k_d = (m, \bar{w})$ with $w\bar{w} = 1 \pmod{m}$. Next, he computes

$$v = \bar{w}c \pmod{m} = \bar{w} \cdot (b_1x_1 + \dots + b_Nx_N) \pmod{m}. \quad (7.6)$$

By definition of \bar{w} and of the b_i s, we have

$$v = (\bar{w}b_1x_1 + \dots + \bar{w}b_Nx_N) \equiv \sum_{i=1}^N x_i a_i \pmod{m}. \quad (7.7)$$

As the sequence a_1, a_2, \dots, a_N is superincreasing, we have

$$a_1 + \dots + a_{N-1} + a_N < a_N + a_N = 2a_N < m$$

and so $v = \sum_{i=1}^N x_i a_i$.

- Now, X knows the integer $\sum_{i=1}^N x_i a_i$ and the superincreasing sequence a_1, a_2, \dots, a_N ; from them he has to reconstruct the integers x_1, x_2, \dots, x_N . This is easily done, in polynomial time, by using the knapsack algorithm for superincreasing sequences.

Remark 7.6.7. Choosing the data in a sufficiently general way, the sequence b_1, \dots, b_N is no more superincreasing, and decrypting illegitimately the message starting with $c = b_1x_1 + \dots + b_Nx_N$ is a computationally hard problem.

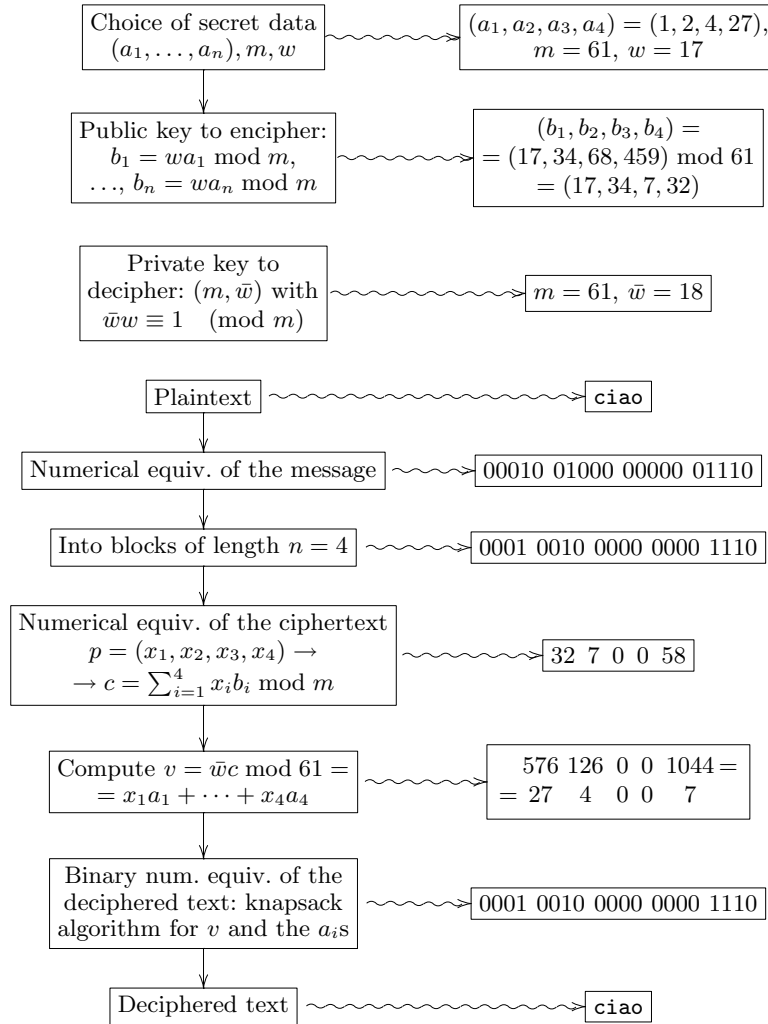
Actually, in 1982 Shamir [52] found a polynomial algorithm that allows one to decrypt the message. The main remark by Shamir is that the knapsack problem to be solved for a sequence of the form b_1, \dots, b_N is not completely general. Indeed b_1, \dots, b_N , even if it is not a superincreasing sequence, may be obtained from a superincreasing sequence a_1, \dots, a_N by means of a very simple transformation.

For these reasons, the cipher just described cannot be considered secure. There are several ways to get around this problem, and quite recently some variations of Merkle–Hellman cipher have been found, that have not yet succumbed to cryptanalysts' attacks. However, the description of these variations goes beyond the scope of this book.

In Table 7.10 on page 350 we show an example of the use of this cipher.

7.7 The *RSA* system

In this section we describe a public key cryptosystem devised by W. Diffie and M. E. Hellman [19], but commonly called *RSA system*, from the names

Table 7.10. Knapsack problem cipher

of those who first implemented it: L. M. Adleman, R. L. Rivest, A. Shamir at M.I.T. (Massachusetts Institute of Technology) [1]. This system, as already remarked, uses a public key that allows enciphering a message but not deciphering it. Each user divulges his enciphering key, so anybody may securely communicate with him. This happens using an enciphering method which is a trapdoor function. The user who divulges the enciphering key *keeps secret an additional piece of information*, by which he alone will be able, by inverting the trapdoor function, to decipher the messages he will receive.

Applications of systems of this kind are innumerable: sending enciphered messages among several users, digital authentication of signatures, access to secure archives or databases or simply services such as credit cards, pay-per-view television programmes, and so forth. We shall not enter into the technical details, leaving the reader with the task of thinking about how the systems we are going to describe can be applied to these situations.

7.7.1 Accessing the *RSA* system

Suppose we want to use the *RSA* system to exchange messages that are to be read only by the intended addressees and not by eavesdroppers. Then we have to join the system, divulging the *enciphering key*, which is a pair of positive integers (n, e) , where n is the product of two large prime numbers p and q we only know, and e must be relatively prime with $\varphi(n) = (p-1)(q-1)$, that is, $\text{GCD}(e, \varphi(n)) = 1$, or $\text{GCD}(e, p-1) = \text{GCD}(e, q-1) = 1$. This pair of integers (n, e) we divulge is kept in a publicly accessible directory.

Remark 7.7.1. How do we proceed *in practice* to find two *large* prime numbers having, for instance, 100 decimal digits? We generate a *random* 100-digit odd number m . *Random generation* of numbers is an interesting topic in mathematics and computer science, upon which we cannot dwell here. Here it suffices to know that there are programs that generate random numbers.

Apply next to m a primality test. If m passes it, then we have found a prime. Otherwise, we apply the primality test to $m+2$. If $m+2$ is not prime either, we test $m+4$, and so on, until a prime number is found. Recall that, by the prime number theorem, the number $\pi(m)$ of prime numbers smaller than m is of the same order as $m/\log m$ (see page 155). A probabilistic rephrasing of the same theorem states that *the frequency with which prime numbers appear near m is $1/\log m$* . So we may expect to have to perform $\mathcal{O}(\log m)$ primality tests before finding the first prime number larger than m . So the number of tests to be performed is polynomial, and so it is feasible. On the other hand, the computational cost of the test itself is usually high.

Returning to the *RSA* system, each user U will act in the same way, that is to say, divulging a pair of integers (n_U, e_U) verifying the same conditions: n_U has to be the product of two prime numbers p_U and q_U that must be large and have to be kept secret, only known to user U , while the second number has to be chosen by U in such a way that $\text{GCD}(e_U, p_U-1) = 1$ and $\text{GCD}(e_U, q_U-1) = 1$.

We emphasise the fact that *the pair (n_U, e_U) is publicly known*, that is, every user who so desires may look it up, while *the factorisation of n_U is not public and is only known to U* . To see how the *RSA* system works, let us consider an example in detail. The general scheme of the procedure is described in Table 7.12 on page 360.

Example 7.7.2. In the public directory, next to the name of each person, their enciphering key will be shown, that is, the pair of integers the user has chosen: for instance,

$$\begin{aligned}
&\text{Ariadne divulges } A = (77, 13), \\
&\text{Beatrix divulges } B = (1003, 3), \\
&\text{Charles divulges } C = (247, 5), \\
&\text{David divulges } D = (703, 7).
\end{aligned} \tag{7.8}$$

The numbers have been chosen by the users according the requisites, as

$$\begin{aligned}
n_A = 77 &= 7 \cdot 11, & \text{GCD}(13, 6) &= \text{GCD}(13, 10) = 1, \\
n_B = 1003 &= 17 \cdot 59, & \text{GCD}(3, 16) &= \text{GCD}(3, 58) = 1, \\
n_C = 247 &= 13 \cdot 19, & \text{GCD}(5, 12) &= \text{GCD}(5, 18) = 1, \\
n_P = 703 &= 19 \cdot 37, & \text{GCD}(7, 18) &= \text{GCD}(7, 36) = 1.
\end{aligned}$$

Notice that in this example we have chosen small numbers, for which it is easy to find the two prime numbers p_U and q_U such that $n_U = p_U q_U$. In general, user U , to be safe, shall use an integer n_U that is the product of two primes of about 100 decimal digits, so the number n_U that is their product and that will be divulged, will have about 200 digits. Notice that in order to implement these kinds of ciphers there is a *crucial* need for many *large* prime numbers. By the way, this fact largely justifies the research about primality tests, as well as the hunt for ever larger prime numbers, which is often covered even by the media. Returning to our example, let us see the next steps, after the publication of the key chosen by each of the users.

7.7.2 Sending a message enciphered with the *RSA* system

User A , Ariadne, has received from user B , Beatrix, a message saying: *Which course do you prefer?* She wants to answer:

algebra

.

To send her answer to Beatrix, Ariadne will have to proceed as follows.

- (1) As a first thing, she transforms each letter of the message into an integer using the 2-digit numerical equivalence, as in Table 7.5 on page 333. Indeed, for the enciphering we are about to describe we shall need the letter-number correspondence assigning two digits to each number associated with a letter. The number sequence corresponding to our message will be

00 11 06 04 01 17 00

.

- (2) Next, Ariadne looks up in the official directory (7.8) the pair of numbers (n_B, e_B) corresponding to Beatrix. Presently she just needs the first number, $n_B = 1003$.
- (3) Now she has to split up the message to be sent into unitary messages so that the integer associated with each unitary message is smaller than

$n_B = 1003$ and relatively prime with 1003. She notices that, having split up the message *algebra* into 2-letter blocks (or digraphs) as follows

$$\boxed{\text{al}} \quad \boxed{\text{ge}} \quad \boxed{\text{br}} \quad \boxed{\text{ax}},$$

the numbers corresponding with the unitary messages are

$$\boxed{0011} \quad \boxed{0604} \quad \boxed{0117} \quad \boxed{0023},$$

that is, the numbers 11, 604, 117, and 23, respectively, which are smaller than 1003 and relatively prime with 1003. Notice that, to find the GCD between these numbers and 1003, Ariadne uses the Euclidean algorithm, as she does not know the prime decomposition of n_B .

Notice further that Ariadne added the letter x at the end of the last unitary message, to make it a digraph. If she had split up the message into 3- rather than 2-letter blocks, she would have found some unitary messages corresponding to integers greater than n_B :

$$\begin{aligned} alg &\longrightarrow 606 < 1003, \\ ebr &\longrightarrow 40117 > 1003, \\ axx &\longrightarrow 2323 > 1003, \end{aligned}$$

which would not have satisfied our requisites.

So the segments of the plaintext will be represented by the following numbers:

$$\boxed{P_1 = 11}, \quad \boxed{P_2 = 604}, \quad \boxed{P_3 = 117}, \quad \boxed{P_4 = 23}.$$

If the numbers P_i were not relatively prime with 1003, we might nevertheless proceed in a way not dissimilar from the one we are about to describe; we are not dwelling on the differences, which the reader may study in Exercise A7.12.

After these operations, we may assume without loss of generality that each unitary message P_i meets the following two conditions:

$$\boxed{P_i < n_B, \quad \text{GCD}(P_i, n_B) = 1.}$$

- (4) Now the actual enciphering of the message to Beatrix begins, in such a way that the latter *may decipher it and be the only person able to do so in a reasonable time*. To encipher the message to be sent to Beatrix, Ariadne raises each P_i to the e_B th power, e_B being the second element of the pair associated with Beatrix.

So the enciphering function is

$$C_B : \mathcal{P} \longrightarrow \mathcal{C}, \quad P \longrightarrow C = P^{e_B} \bmod n_B.$$

Thus, the enciphered message Ariadne will have to send will consist of the following numbers:

$$\begin{aligned}
C_1 &= P_1^{e_B} \bmod n_B = 11^3 \bmod 1003 = 328, \\
C_2 &= P_2^{e_B} \bmod n_B = 604^3 \bmod 1003 = 797, \\
C_3 &= P_3^{e_B} \bmod n_B = 117^3 \bmod 1003 = 825, \\
C_4 &= P_4^{e_B} \bmod n_B = 23^3 \bmod 1003 = 131.
\end{aligned}$$

Then Beatrix receives the following message:

$$\boxed{C_1 = 328}, \boxed{C_2 = 797}, \boxed{C_3 = 825}, \boxed{C_4 = 131},$$

consisting of the unitary messages C_i , $i = 1, \dots, 4$. Now Ariadne has carried out her task: she has sent Beatrix the enciphered message. Now Beatrix will have to decipher it.

7.7.3 Deciphering a message enciphered with the *RSA* system

Beatrix has received the message

$$\boxed{C_1 = 328}, \boxed{C_2 = 797}, \boxed{C_3 = 825}, \boxed{C_4 = 131},$$

and has to decipher it, that is, for every C_i has to find the original message P_i , knowing that

$$C_i = P_i^{e_B} \bmod n_B.$$

So Beatrix has to determine the deciphering function

$$D_B : \mathcal{C} \longrightarrow \mathcal{P}$$

such that $D_B(E_B(P_i)) = P_i$ for all i . How can she find it? A priori it would seem that, to solve this problem, Beatrix would have to find a discrete logarithm which, as we have remarked, is computationally quite hard. However, we have already remarked at the beginning that Beatrix actually has *an additional piece of information enabling her to decipher the message she has received* without difficulty. Let us see which piece of information she has and how she uses it. First of all, Beatrix determines d_B , with

$$1 \leq d_B < \varphi(n_B) = (p_B - 1)(q_B - 1),$$

such that d_B is a solution of the following congruence

$$\boxed{e_B d_B \equiv 1 \pmod{\varphi(n_B)}}. \quad (7.9)$$

Such a solution exists and is unique, as $\text{GCD}(e_B, \varphi(n_B)) = 1$.

In our case, $n_B = 1003$, and Beatrix knows $\varphi(1003)$ because she knows that $1003 = 17 \cdot 59$. So $\varphi(1003) = 16 \cdot 58 = 928$. Then the solution d_B of the congruence (7.9), that is of $3x \equiv 1 \pmod{928}$, is

$$\boxed{d_B = 619}.$$

This number is truly to be framed, because, as we shall shortly see, Beatrix is the only person who can decipher the message because she is the only one who, knowing the factorisation of 1003, is able to compute its Euler function and so d_B , which is Beatrix's *private key* to decipher the messages sent to her. Let us what she does to decipher the messages C_i .

Beatrix raises each C_i to the power $d_B = 619$, that is, computes

$$328^{619}, 797^{619}, 825^{619}, 131^{619}.$$

The exponent 619 is large, but we know how to proceed in situations like this (see § 3.3.1). The number 619 is written in base 2, that is

$$619 = (1001101011)_2 = 512 + 64 + 32 + 8 + 2 + 1.$$

So we have

$$C_i^{619} = C_i^{512} \cdot C_i^{64} \cdot C_i^{32} \cdot C_i^8 \cdot C_i^2 \cdot C_i^1$$

and the powers are easily computed according to the following table, for $C_1 = 328$:

k	$C_1^k \bmod 1003$
1	$\boxed{328}$
2	$328^2 \bmod 1003 = \boxed{263}$
$2^2 = 4$	$263^2 \bmod 1003 = 965$
$2^3 = 8$	$965^2 \bmod 1003 = \boxed{441}$
$2^4 = 16$	$441^2 \bmod 1003 = 902$
$2^5 = 32$	$902^2 \bmod 1003 = \boxed{171}$
$2^6 = 64$	$171^2 \bmod 1003 = \boxed{154}$
$2^7 = 128$	$154^2 \bmod 1003 = 647$
$2^8 = 256$	$647^2 \bmod 1003 = 358$
$2^9 = 512$	$358^2 \bmod 1003 = \boxed{783}$

where we have framed the factors to be multiplied. From the table we may easily see that

$$\begin{aligned} 328^{619} &= 328^{512} \cdot 328^{64} \cdot 328^{32} \cdot 328^8 \cdot 328^2 \cdot 328^1 \equiv \\ &\equiv 783 \cdot 154 \cdot 171 \cdot 441 \cdot 263 \cdot 328 \equiv 11 \pmod{1003}. \end{aligned}$$

Notice that, by raising $C_1 = 328$ to the exponent $d_B = 619$, we have obtained the number $11 = P_1$, which is the number corresponding to the first part of the original message.

We do the same for the three other message segments C_2 , C_3 and C_4 . So we get Table 7.11 on page 356 where, as above, we have framed the factors to be multiplied. In conclusion,

Table 7.11. Powers of the numbers C_i

k	$C_2^k \bmod 1003$	$C_3^k \bmod 1003$	$C_4^k \bmod 1003$
1	797	825	131
2	$797^2 \equiv 310$	$825^2 \equiv 591$	$131^2 \equiv 110$
4	$310^2 \equiv 815$	$591^2 \equiv 237$	$110^2 \equiv 64$
8	$815^2 \equiv 239$	$237^2 \equiv 1$	$64^2 \equiv 84$
16	$239^2 \equiv 953$	1	$84^2 \equiv 35$
32	$953^2 \equiv 494$	1	$35^2 \equiv 222$
64	$494^2 \equiv 307$	1	$222^2 \equiv 137$
128	$307^2 \equiv 970$	1	$137^2 \equiv 715$
256	$970^2 \equiv 86$	1	$715^2 \equiv 698$
512	$86^2 \equiv 375$	1	$698^2 \equiv 749$

$$\begin{aligned}
328^{619} \bmod 1003 &= 11, \\
797^{619} \bmod 1003 &= 604, \\
825^{619} \bmod 1003 &= 117, \\
131^{619} \bmod 1003 &= 23.
\end{aligned}$$

These are to be seen as 4-digit numbers:

$$0011, \quad 0604, \quad 0117, \quad 0023,$$

and correspond to the four original message segments, which were digraphs. So we have to split each of them into two parts, each of which represents a letter. So Beatrix, using Table 7.5 on page 333, finds

$$\begin{array}{cccccccc}
00 & 11 & 06 & 04 & 01 & 17 & 00 & 23 \\
a & l & g & e & b & r & a & x
\end{array}$$

and gets to know the course Ariadne likes best.

With did this work? That is, why raising C_i to the exponent d_B we get back P_i such that $C_i = P_i^{e_B}$? In other words, why is

$$D_B : \mathcal{C} \longrightarrow \mathcal{P}, \quad C_i \longrightarrow C_i^{d_B}$$

the deciphering function? Here follows the reason.

7.7.4 Why did it work?

First of all, notice that the congruence (7.9) has exactly one solution modulo $\varphi(n_B)$, because the coefficient e_B is such that

$$\text{GCD}(e_B, p_B - 1) = 1, \quad \text{GCD}(e_B, q_B - 1) = 1,$$

so also $\text{GCD}(e_B, (p_B - 1)(q_B - 1)) = 1$.

Remark 7.7.3. Beatrix is the only person able to solve congruence (7.9), because she is the only one to know the Euler function $\varphi(n_B) = (p_B - 1)(q_B - 1)$, as she knows the prime factors p_B and q_B of n_B . In fact, notice that, as p_B and q_B are *large* prime numbers, factoring n_B normally takes a very long time. So, in fact, Beatrix is the only one to know this factorisation.

Actually, one might doubt that knowing $\varphi(n_B)$ is *equivalent* to knowing the prime factors p_B and q_B . Of course, whoever knows these factors knows $\varphi(n_B)$ too. But, is it possible to know $\varphi(n_B)$ without knowing p_B and q_B ? The answer is no: if $\varphi(n_B)$ is known, then p_B and q_B can be reconstructed immediately, in polynomial time. The easy proof is left as an exercise (see Exercise A7.10).

Notice now that d_B is actually the *private key* allowing Beatrix to decipher the message. Indeed, setting $P = P_i$ and $C = C_i$, we have

$$P \equiv C^{d_B} \pmod{n_B},$$

as

$$C^{d_B} \equiv (P^{e_B})^{d_B} = P^{e_B d_B} \pmod{n_B}.$$

On the other hand, $e_B d_B \equiv 1 \pmod{\varphi(n_B)}$ implies that $e_B d_B - 1$ is a multiple of $\varphi(n_B)$, that is $e_B d_B = 1 + \varphi(n_B)k$ for some k . So,

$$P^{e_B d_B} = P^{1 + \varphi(n_B) \cdot k} = P \cdot (P^{\varphi(n_B)})^k.$$

As $\text{GCD}(P, n_B) = 1$, by Euler's theorem we have $P^{\varphi(n_B)} \equiv 1 \pmod{n_B}$; hence

$$P^{e_B d_B} \equiv P \pmod{n_B}.$$

So,

$$P \equiv C^{d_B} \pmod{n_B}.$$

As Ariadne has chosen $P < n_B$, there is no ambiguity in determining the number congruent to C^{d_B} modulo n_B : it is the only such number between 0 and $n_B - 1$. Once this P is found, Beatrix can read Ariadne's message.

Remark 7.7.4. We have said that the unitary message has to be smaller than n_B . We have just explained the reason of this request. An example will illustrate the need for it. Consider the message

no

to be sent to Ariadne, whose pair is $(n_A = 77, e_A = 13)$. We opt to consider the whole word *no* as unitary message (digraph). We proceed as above:

- (1) transform the message into a number by associating to each letter its numerical equivalent. The associated number is found to be

$$1314;$$

notice that 1314 is greater than $n_A = 77$;

(2) raise 1314 to the power $e_A = 13$; we have

$$C_1 = 1314^{13} \equiv 26 \pmod{77};$$

(3) Ariadne receives the message

26.

To decipher this message Ariadne uses her private key, which is $d_A = 37$, the solution of the congruence

$$13d_A \equiv 1 \pmod{\varphi(77)}, \quad \text{that is} \quad 13d_A \equiv 1 \pmod{60}.$$

Raising 26 to the power 37, Ariadne gets $5 \pmod{77}$, which she interprets as

f

so she cannot reconstruct the message she was sent. Also notice that $1314 \pmod{77} = 5$.

So, if we do not request for *the unitary message to be smaller than n_A* , that is, than the first element of the pair of numbers published by the addressee, it becomes *impossible* to define the deciphering transformation.

Remark 7.7.5. We have seen that, in order to send the message *algebra* to Beatrix, Ariadne split it up into digraphs. She had to do so by trial and error, verifying all the numbers corresponding to the single digraphs to be smaller than n_B and relatively prime with it.

However, there is a better way of choosing how to split up the message: rather than splitting the original message *algebra*, it is more convenient to split the *numerical* message obtained by associating a 2-digit number with each letter. In this way there is a natural way of splitting it, as follows.

After transforming the message into a sequence of 2-digit numbers, consider the number consisting of the sequence of all the digits, which will be called *numerical message*. Split it up into k -digit blocks, where

$$k = (\text{number of digits of } n_B) - 1.$$

In this way, *without even having to examine the message*, each unitary numerical message is smaller than n_B . What's more, everybody concerned knows n_B and knows that the sender will split up the message into blocks like this.

Let us illustrate this new method with an example.

Example 7.7.6. Suppose Ariadne, user A , sends Beatrix, user B , the message

come here

Then:

(1) first of all Ariadne transforms the message, ignoring spaces, into the sequence of 2-digits numbers

02 14 12 04 07 04 17 04

which will be written as

0214120407041704.

There is no ambiguity, as we know that the number associated with each letter of the original message consists of two digits. This is the *numerical message*;

- (2) as the number $n_B = 1003$ has 4 digits, A splits the numerical message into blocks of length $4 - 1 = 3$, that is, into trigraphs, as follows:

$$\boxed{021 \quad 412 \quad 040 \quad 704 \quad 170 \quad 423}.$$

Notice that A has added 23, corresponding to the letter x , at the end of the message, so all unitary numerical messages consist of three digits. In this way, she has split the numerical message into unitary numerical messages that are trigraphs, and each unitary message is certainly smaller than $n_B = 1003$. Notice that this partitioning is not a partition of the original message *come here*, as the 3-digit unitary numerical blocks do not correspond to any letter group. As we shall see, this will not create any difficulty.

We still have to check that each P_i is relatively prime with 1003. It is easily verified that the only exception is 170. However, we have no reason to worry: keeping in mind Exercise A7.12, we may go on.

So, the unitary numerical messages are

$$P_1 = 21, \quad P_2 = 412, \quad P_3 = 40, \quad P_4 = 704, \quad P_5 = 170, \quad P_6 = 423.$$

Notice that the operations carried out so far do not amount to any enciphering, as transforming a message into a numerical sequence is a standard operation, and so is partitioning it into 3-digit blocks according to the value n_B , which is known to everybody;

- (3) the enciphered message will be represented by the following numbers:

$$\begin{aligned} C_1 &= P_1^{e_B} \bmod n_B = 21^3 \bmod 1003 = 234, \\ C_2 &= P_2^{e_B} \bmod n_B = 412^3 \bmod 1003 = 353, \\ C_3 &= P_3^{e_B} \bmod n_B = 40^3 \bmod 1003 = 811, \\ C_4 &= P_4^{e_B} \bmod n_B = 704^3 \bmod 1003 = 54, \\ C_5 &= P_5^{e_B} \bmod n_B = 170^3 \bmod 1003 = 306, \\ C_6 &= P_6^{e_B} \bmod n_B = 423^3 \bmod 1003 = 587. \end{aligned}$$

So Beatrix receives the following sequence of unitary messages:

$$\boxed{C_1 = 234, C_2 = 353, C_3 = 811, C_4 = 54, C_5 = 306, C_6 = 587}.$$

To decipher it, she raises each C_i to her private key $d_B = 619$, that is, she computes

$$234^{619}, 353^{619}, 811^{619}, 54^{619}, 306^{619}, 587^{619}.$$

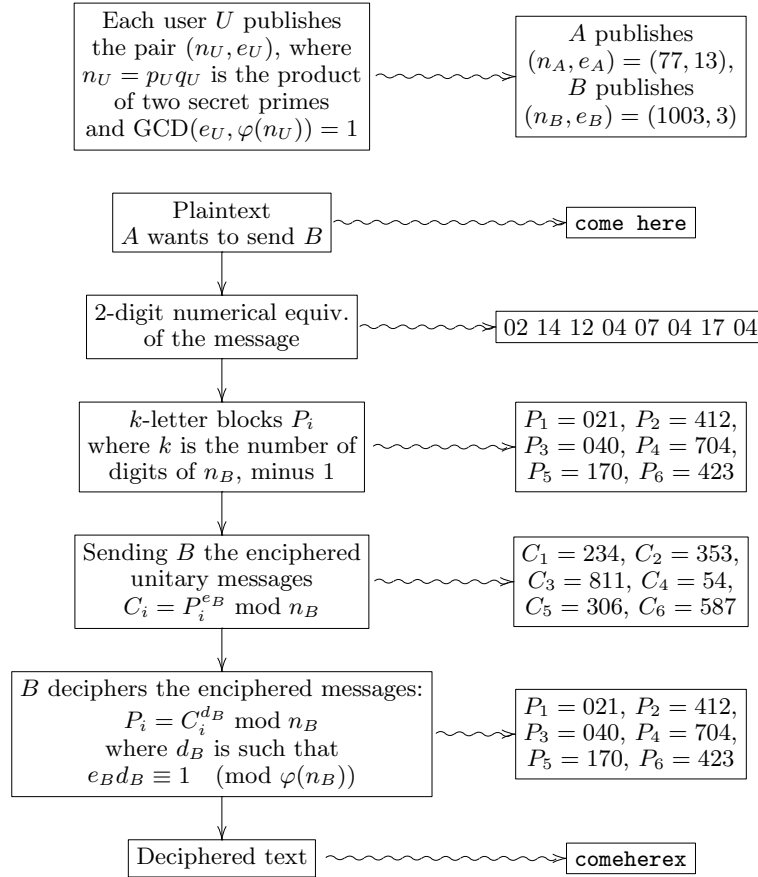
In this way, Beatrix finds

$$\begin{aligned} 234^{619} \bmod 1003 &= 21, & 353^{619} \bmod 1003 &= 412, \\ 811^{619} \bmod 1003 &= 40, & 54^{619} \bmod 1003 &= 704, \\ 306^{619} \bmod 1003 &= 170, & 587^{619} \bmod 1003 &= 423. \end{aligned}$$

This 3-digit blocks (completed with a leading zero where necessary), regrouped in twos, give

$$02, 14, 12, 04, 07, 04, 17, 04, 23$$

and now Beatrix can read the message *comeherex*, which she understand as *come here*. This example is shown in Table 7.12 on page 360.

Table 7.12. Example of use of *RSA* system

7.7.5 Authentication of signatures with the *RSA* system

The *RSA* system allows one to solve of an important problem which is more and more relevant in this era of telecommunications: the problem of *digitally authenticating a signature*.

If Beatrix receives a message from a person signing herself Ariadne, how can she be sure the sender was actually Ariadne? The certainty may be achieved as follows.

Ariadne writes her message P_1 , putting at the end her signature F ; to authenticate the signature, Ariadne adds after the message P_1 the message

$$P_2 = F^{d_A} \bmod n_A,$$

where d_A is her *private key*, that is, the key known only to her, because only she knows the factorisation of her public key n_A . Then she sends Beatrix the message P consisting of the two messages P_1 and P_2 as usual, that is, raising P_1 and P_2 to the power e_B and reducing them modulo n_B .

On receiving the message, Beatrix reads it using her private key d_B . Deciphering message P_1 , she learns that the message was sent by Ariadne, because the message is signed with Ariadne's signature F . But was really Ariadne, and not someone else, who used that signature? Here the section P_2 of the message gives an answer. Indeed, it consists of some undecipherable characters, which nevertheless contain the *proof of the authenticity of the signature*.

Now Beatrix, to verify the authenticity, has to proceed as follows. To decipher P_2 she cannot use *her private key* d_B , which would be useless, as the original message F was enciphered raising it not to the power e_B but to d_A . Instead, Beatrix uses Ariadne's public key e_A . In this way *she obtains Ariadne's signature* F , because

$$P_2^{e_A} \equiv (F^{d_A})^{e_A} = F^{d_A e_A} \equiv F \pmod{n_A}.$$

This signature has to be authentic, as only Ariadne knows her private key. If what appeared were not Ariadne's signature F , the message would have been a fake. Basically, to authenticate a signature, the *sender* uses *her private key*, rather than the *addressee*.

Example 7.7.7. Recall that Ariadne published the pair $(n_A = 77, e_A = 13)$. As $77 = 11 \cdot 7$, Ariadne's private key is $d_A = 37$, because $37 \cdot 13 \equiv 1 \pmod{60}$, where $60 = \varphi(77)$. In sending a message to Beatrix, Ariadne authenticates her signature, which we assume to be $F = 5$, raising 5 to the power d_A modulo n_A :

$$5^{37} \bmod 77 = 47.$$

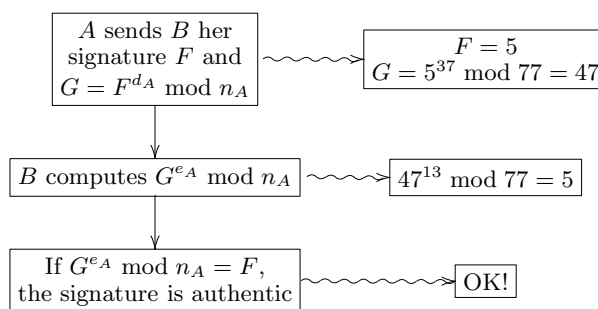
Beatrix verifies the authenticity of the signature by raising 47 to the power $e_A = 13$ modulo n_A :

$$47^{13} \bmod 77 = 5,$$

that is, she gets again Ariadne's signature. So Beatrix is sure that the message's author is Ariadne.

The previous example is summarised in Table 7.13.

Table 7.13. Authentication of a signature with the RSA system



7.7.6 A remark about the security of *RSA* system

The security of *RSA* system lies in the fact that, as already emphasised several times in earlier chapters, so far there is no efficient algorithm to factor large numbers. If A sends B a message C , an unauthorised eavesdropper who tried to decrypt it should be able to find the factorisation of n_B . To find it, when n_B is the product of two 60-digit primes, even using the most advanced algorithms and the fastest computers, would require several months, if not years. The situation is even more unfeasible if we choose primes with 100 or more digits: in this case factoring n is, in practice, impossible. However, if this is true *in general*, it is not *always* so, as the following episode shows.

In August 1977 the three inventors of the public key *RSA* cryptosystem, Rivest, Shamir and Adleman, at MIT, challenged from Martin Gardner's column *Mathematical Games* the readers of *Scientific American* to decrypt a message corresponding to a 129-digit number, an operation they believed to require billions years. They offered a reward of \$100 to whomever found the solution.

We are not going to give all the details of Rivest, Shamir and Adleman's problem. Suffice it to say that, in order to decrypt their original message it was necessary to factor the number

$$N = 1143816257578886766923577997614661201021829672124236256256184293 \\ 5706935245733897830597123563958705058989075147599290026879543541,$$

which had been published together with the number $e = 9007$. Basically, (N, e) was the *public key* of Rivest, Shamir and Adleman.

To ensure that the message came from the MIT team, the following *digital signature* was added, using the private key of the algorithm, that is the number d such that $ed \equiv 1 \pmod{N}$:

$$1671786115038084424601527138916839824543690103235831121783503844 \\ 6929062655448792237114490509578608655662496577974840004057020373.$$

Raising this number to the power 9007, then reducing it modulo N , one obtained the number

$$0609181920001915122205180023091419001 \\ 5140500082114041805040004151212011819$$

corresponding, in Rivest, Shamir and Adleman's cipher, to the sentence:

First solver wins one hundred dollars,

which guaranteed that the message really came from MIT.

Seventeen years later, the Dutch mathematician Arjen K. Lenstra, together with a team of hundreds, in just 8 months managed to find the solution. The technique used in tackling the problem is the so-called *multiple polynomial quadratic sieve*, a technique that allows to split up the task into several smaller subtasks. Using this sieve, the possible factors are found among millions of candidates. To organise the work, Lenstra needed hundreds of collaborators all over the world and involved thousands of computers, the whole enterprise being coordinated via the Internet.

The results of this collective effort were sent Lenstra: two days' computations with a supercomputer produced a 64-digit and a 65-digit factor. This allowed Lenstra to decrypt the message by Rivest, Shamir and Adleman.

Are you curious to know what the message said? It said:

the magic words are squeamish ossifrage.

The three scientists themselves said that it was a meaningless sentence: they would never have supposed, when they wrote it, that it would someday emerge. What had seemed an impossible challenge, seventeen years later turned out to be within the grasp of the most advanced researchers. The conclusion is that cryptography is still, in several regards, an *experimental science*. It still relies on several conjectures, such as Diffie–Hellman hypothesis, we shall deal with shortly, which might be, if not completely contradicted, at least quite diminished when new algorithms are invented that, at least in many cases, do a good work to elude them. So, when there are no *theorems* telling us whether a given cryptographic procedure is secure, it is convenient to be careful rather than doing as if it were certainly so. A system that today is believed to be secure might not be so tomorrow, as we shall see in Chapter 9.

7.8 Variants of *RSA* system and beyond

We are now going to describe some cryptosystems, the first of which is a variant of *RSA* system. Its security relies on the problem of computing discrete logarithms.

7.8.1 Exchanging private keys

The *RSA* system, or rather a slight modification of it, allows two users to exchange a private key with which, *independently of the public key system*, they can exchange enciphered messages using one of the classic methods discussed at the beginning of this chapter.

Let us modify the *RSA* system as follows. Choose a very large prime number p , which is divulged, and work in the field \mathbb{Z}_p . Actually, we might work in any finite field \mathbb{F}_q , but we shall limit ourselves to p -element fields. Choose next a non-zero element $g \in \mathbb{Z}_p$, which is divulged too. The most convenient choice, to use in the best way the system's resources, would be to choose as g a generator of the multiplicative group of the field \mathbb{Z}_p . However, this is not strictly necessary.

Moreover, each user U chooses his *private key* e_U , which is a positive number smaller than $p - 1$, and divulges $g^{e_U} \in \mathbb{Z}_p$, that is, the positive number $X_U = g^{e_U} \bmod p$. Notice that, from X_U , it is not possible to reconstruct in a reasonable time U 's private key e_U . Indeed, this would imply finding a *discrete logarithm* which, as we know, requires in general an exponential time.

Let us see now how two users A and B may proceed to exchange a private key. There is a very simple method: A and B may agree to use as a private key $g^{e_A e_B} \in \mathbb{Z}_p$, that is, the number

$$X_{AB} = g^{e_A e_B} \bmod p.$$

Indeed, both A and B can compute X_{AB} in polynomial time. For instance, A knows X_B , which is public. Moreover, she knows her private key e_A . So she computes

$X_B^{e_A} = (g^{e_B})^{e_A} \bmod p = X_{AB}$. Similarly, B knows X_A and e_B , so can compute $X_A^{e_B} = (g^{e_A})^{e_B} \bmod p = X_{AB}$.

On the other hand, an *eavesdropper* C will find it hard to compute X_{AB} . In fact, he knows X_A and X_B , but how can he reconstruct X_{AB} from them? In order to do so, he probably should find first e_A and e_B , and compute next $g^{e_A e_B}$, which would finally allow him to figure out X_{AB} . But in order to find e_A and e_B , C should compute some discrete logarithms, which is computationally unfeasible.

But, are we certain that to compute X_{AB} knowing X_A and X_B it is necessary to compute some discrete logarithms? In other words, are we sure that in order to compute $g^{e_A e_B}$ knowing g^{e_A} and g^{e_B} it is necessary to know e_A and e_B ? So far, nobody has proved nor disproved this fact. Nevertheless, it is conjectured that the complexity of computing $g^{e_A e_B}$ knowing g^{e_A} and g^{e_B} is equal to that of finding discrete logarithms: this is the so-called *Diffie–Hellman hypothesis*. On this hypothesis the security of this method of exchange of private keys is based.

Example 7.8.1. Assume $p = 19$ and $g = 2$ have been divulged. Let $e_A = 16$ and $e_B = 11$ be the private keys of A and B , respectively. Then A and B *publish* the values

$$X_A = 2^{16} \bmod 19 = 5, \quad X_B = 2^{11} \bmod 19 = 15,$$

respectively. The common key A and B will use to exchange messages is $X_{AB} = g^{e_A e_B} \bmod 19$. A will compute it as follows:

$$X_{AB} = X_B^{e_A} \bmod 19 = 15^{16} \bmod 19 = 6.$$

Clearly B gets the same result by computing

$$X_A^{e_B} \bmod 19 = 5^{11} \bmod 19 = 6.$$

Notice that A and B can now use the key 6 to exchange messages enciphered, for instance, using a Caesar cipher operating on 26 letters: 6 might be the enciphering key, that is, the number of positions the letters are shifted in Caesar cipher.

7.8.2 ElGamal cryptosystem

Fix a large finite field \mathbb{F}_q (we may well take \mathbb{Z}_p , for a large p) and an element $g \in \mathbb{F}_q^*$ (preferably, but not necessarily, a generator of \mathbb{F}_q^*). We shall assume that the numerical equivalents of the messages are in \mathbb{F}_q .

Each user A has a public key and a private key: the private key is an integer $a = a_A$, randomly chosen by A ($0 < a < q - 1$), while the public key is $g^a \in \mathbb{F}_q$.

Assume B wants to send A a message P . Then B proceeds as follows:

- B randomly chooses an integer $k < q$;
- he computes g^k in \mathbb{F}_q ;
- he computes g^{ak} in \mathbb{F}_q ;
- he multiplies the message P by g^{ak} in \mathbb{F}_q ;
- he sends A the pair $(y_1 = g^k, y_2 = P \cdot g^{ak})$.

Notice that in order to compute g^{ak} it is not necessary to know A 's private key; it suffices to know g^a , as $g^{ak} = (g^a)^k$.

On receiving the pair (y_1, y_2) , A , who knows a , which is her own private key, can discover the message P by raising y_1 to the exponent a and dividing y_2 by the result found. Indeed,

$$y_2 \cdot ((y_1)^a)^{-1} = P \cdot g^{ak} \cdot ((g^k)^a)^{-1} = P.$$

Somebody who could solve the discrete logarithm problem could violate the cryptosystem by determining the private key a from the knowledge of g^a . In theory, it could be possible to obtain g^{ak} knowing g^a and g^k (and so to arrive at the message P), but here too, as already said in § 7.8.1, it is conjectured that solving this kind of problem without solving the problem of computing discrete logarithms is impossible.

7.8.3 Zero-knowledge proof: or, persuading that a result is known without revealing its content nor its proof

Suppose Paul has found a very important formula: he wants to persuade a colleague he has found it, *but without giving him any indication about the formula itself nor about the way he has proved it*. Is it possible? This kind of communication is said to be a *zero-knowledge* protocol, that is to say, it is a communication that does not transmit any information that could give away the formula or its proof, but lets the addressee know we actually have it. It looks like an impossible feat. However, we shall see that it is possible. Let us see an example to illustrate how to proceed.

Let G be a finite group with N elements, and let b and y be two elements of G . Suppose Paul has found a discrete logarithm for y in base b , that is, he has determined a positive integer x such that

$$b^x = y.$$

His friend Sylvia is sceptical: Paul wants to convince her he knows x *without telling her* x . Assume Sylvia knows the order N of the group G (the case in which Sylvia does not know N can also be dealt with, but we shall not do so). They may proceed as follows:

- (1) Paul generates a random positive integer $e < N$ and sends Sylvia

$$b' = b^e;$$

- (2) Sylvia tosses a coin: if it shows heads, Paul must disclose e to Sylvia and she checks whether actually $b' = b^e$;
- (3) if the coin shows tails, then Paul must disclose the positive integer $x + e \bmod N$. As $b^x = y$ and $b^e = b'$, we have $b^{x+e} = yb'$. Sylvia will check that the number has the required property (notice that Sylvia knows both y and b').

The three steps are repeated (and so there will be a *new* choice of a random integer e , a new coin toss, and so on), until Sylvia is convinced that Paul has actually found the discrete logarithm of y .

How can she be convinced? If Paul did not really know the discrete logarithm of y and were cheating, he would be able to *answer just one of the two possible questions*. If the coin comes up as heads he certainly may disclose e , but if it comes up as tails, how can he disclose $x + e \bmod N$ without knowing x ? He might try to elude the problem by sending, in step (1), $b' = b^e/y$ rather than b^e : so, if tails shows

up, he may reveal $e = (e - x) + x$ (which he can easily do). But in this case he would be exposed if the coin shows heads: indeed, in this case he should reveal $e - x$, and how could he without knowing x ?

By iterating the procedure a sufficient number of times, sooner or later Sylvia *will be persuaded* that Paul actually knows what he claims to know.

So Paul manages to *prove* Sylvia that he knows the discrete logarithm x of y without explicitly exhibiting x , and so his secret remains his own.

7.8.4 Historical note

This *challenge* scheme calls to mind the challenges that took place centuries ago. In the 16th century a mathematician's ability was demonstrated through "public challenges": these scientific duels were actual tournaments with witnesses, judges, referees and so on. In these challenges fame and money were at stake. For this reason, the most important discoveries were kept *jealously secret*. So, when a mathematician came in possession of a new discovery, he sent a *cartello di matematica disfida* (public mathematical challenge), in which he claimed to be able to solve a class of problems and proposed in turn some of them, and the "contenders" engaged in proposing and solving such problems.

Among the most famous challenges, there were those between Dal Fior and Tartaglia (Nicolò Fontana): the problems presented by Dal Fior can be reduced to solving equations of the form $x^3 + px = q$, which Dal Fior could solve because their solution was transmitted him by his teacher Scipione Dal Ferro before dying. Tartaglia proposed a series of problems reducible to the solution of equations of the form $x^3 + mx^2 = q$, which he could solve. It happened that, even without knowing the general formula for the equations in possession of Dal Fior, Tartaglia managed to find it in time to solve all the problems, while Dal Fior could not solve any.

Another renowned challenge was the one between Ferrari and Tartaglia in 1548: Tartaglia in 1539 had given Cardano the solution of a class of third degree equations (the *casus non irriducibilis*), making him promise he would not divulge it. In 1545 Cardano published his work "Ars Magna" in which, violating his promises, he gave the formula to solve cubic equations. Tartaglia took offence and Cardano's pupil Ferrari challenged him in another famous confrontation.

7.9 Cryptography and elliptic curves

So far we have only described the development of several classic and modern cryptographic methods, all based on algebraic, and mostly arithmetic, ideas. In other words, these methods rely on properties of numbers or their congruence classes.

In this section we are going to discuss some new frontiers recently opened to cryptography, especially for what regards the security and the prevention of cryptanalysis. This is due to the interaction of classic algebra and arithmetic with ideas and notions from geometry, and in particular from the study of certain plane curves called *elliptic curves*.

7.9.1 Cryptography in a group

Before going on, we give explicitly a remark that, in an implicit form, we have already mentioned elsewhere in this chapter. To fix ideas, consider the exchange of private keys through the *RSA* system, described in the previous section. It relies on exponentiation in \mathbb{Z}_p , with p a prime number. Its easy execution is due to the fact that exponentiating in \mathbb{Z}_p is computationally easy, that is, requires a polynomial time. Its security, on the other hand, depends on the fact that finding discrete logarithms in \mathbb{Z}_p is apparently much harder computationally. More precisely, as we have seen, the Diffie–Hellman hypothesis is relevant here (see page 364).

On the other hand, the theoretical basis of this cryptographic system works *with no changes* if, rather than in the multiplicative group \mathbb{Z}_p^* , we work in any other finite group G . Leaving the description of the details of the scheme as an exercise, we just remark that actually, to put in practice the theory, and so to implement a cryptosystem to exchange private keys based on exponentiation in an arbitrary finite group G , we must ask that:

- it is possible to perform computations in G , that is, it is necessary that G is given not only in a theoretical way, but operatively, in such a way that we can actually *work* with its elements;
- exponentiating in G is easy, that is, requires, for instance, a polynomial computational cost;
- determining discrete logarithms in G is computationally much harder, for instance exponential, and that in G the Diffie–Hellman hypothesis holds, that is, for a randomly chosen element $g \in G$ and for $a, b \in \mathbb{Z}$, computing g^{ab} knowing g^a and g^b has *the same computational difficulty* as determining discrete logarithms in G .

For instance, if \mathbb{F}_q , $q = p^f$, with p a prime number, is a finite field and $G = \mathbb{F}_q^*$ is its multiplicative group, then G has these properties, as:

- G can be described concretely as an extension of \mathbb{Z}_p . Some of the examples in Chapter 5 show how to describe its elements;
- exponentiating is easy in group G , that is, it requires a polynomial computational cost (see § 5.1.14);
- just like in \mathbb{Z}_p , it is conjectured that determining discrete logarithms in G has at least an exponential cost, and that in G the Diffie–Hellman hypothesis holds: indeed, it is clear that if it holds in \mathbb{Z}_p then it holds in \mathbb{F}_q^* too.

So we may use in cryptography, and it is actually used, the multiplicative group of a finite field \mathbb{F}_q rather than that of \mathbb{Z}_p with p a large prime. This yields remarkable advantages. For instance, we may use fields of the form \mathbb{F}_{2^n} of characteristic 2, which are very suitable for a computational approach because their elements can be described as n -tuples of 0s and 1s. Moreover, by choosing a *large* n , \mathbb{F}_{2^n} becomes in turn *large* very quickly, removing the

need for a large prime p to construct \mathbb{Z}_p . Unfortunately, choosing \mathbb{F}_{2^n} makes life easier for cryptanalysis. Indeed, recently, in 1984, D. Coppersmith found efficient algorithms to compute discrete logarithms in these fields (see [14], [44]).

So, which groups may we use to do cryptography? We would like groups *quite similar* to \mathbb{F}_q^* , which would make them *familiar-looking* and, most important, *computationally easy to use*. At the same time, we would have many of them, to be able to choose among them, perhaps change them frequently, to avoid too easy a cryptanalysis.

Here geometry lends us a helping hand. Let us explore the ideas that lead to considering elliptic curves.

7.9.2 Algebraic curves in a numerical affine plane

Rather than considering specifically \mathbb{F}_q , consider an arbitrary field \mathbb{K} . So we may define the *numerical affine plane* $\mathbb{A}_{\mathbb{K}}^2$ with coordinates on this field (see [51]). Basically, this is just $\mathbb{K} \times \mathbb{K}$. This terminology is not surprising, and has already been used before. In fact, just consider the case $\mathbb{K} = \mathbb{R}$, leading to the usual plane $\mathbb{A}_{\mathbb{R}}^2$ with cartesian coordinates (x, y) .

In the affine plane $\mathbb{A}_{\mathbb{K}}^2$ we may *do geometry* exactly as in the real cartesian plane. For instance, we may consider *algebraic curves*. These are subsets of $\mathbb{A}_{\mathbb{K}}^2$ defined by an equation of the form

$$f(x, y) = 0, \quad (7.10)$$

where $f(x, y)$ is a polynomial with coefficients in \mathbb{K} , which we assume to be non-constant and without repeated factors. The curve defined by (7.10) is the set of points $(u, v) \in \mathbb{A}_{\mathbb{K}}^2$ such that $f(u, v) = 0$. Clearly, substituting the polynomial $kf(x, y)$ for $f(x, y)$, where $k \in \mathbb{K}^*$, we obtain the same curve.

The curve defined by Equation (7.10) is said to be *irreducible* if the polynomial $f(x, y)$ is irreducible over \mathbb{K} . Notice that this notion *depends* on the field \mathbb{K} , because, as we know, a polynomial may be irreducible over \mathbb{K} but not over an extension of \mathbb{K} .

Example 7.9.1. Consider the curve in $\mathbb{A}_{\mathbb{R}}^2$ having equation $x^2 + y^2 = 0$. It is irreducible because such is the polynomial $x^2 + y^2$ over \mathbb{R} . On the contrary, the curve in $\mathbb{A}_{\mathbb{C}}^2$ with the same equation is reducible because we have $x^2 + y^2 = (x + iy)(x - iy)$ over \mathbb{C} .

Notice that the curve in $\mathbb{A}_{\mathbb{R}}^2$ having equation $x^2 + y^2 = 0$ consists of the single point having coordinates $(0, 0)$. So the definition must be studied carefully: the notion of a curve includes sets which do not always correspond to the intuitive idea of a curve as the reader may picture it!

If $f(x, y)$ has degree d , we say that d is the *degree* of the curve of equation (7.10). The curves of degree 1, which are clearly irreducible (see Exercise A7.13), are called *lines*, those of degree 2 *conic* curves, those of degree 3 *cubic*, those of degree 4 *quartic* and so forth.

7.9.3 Lines and rational curves

The x -axis, which has equation $y = 0$, may be identified in a natural way with the field \mathbb{K} , as it consists of all points $(x, 0)$, with x ranging in \mathbb{K} . An analogous remark can be made about the y -axis, which has equation $x = 0$.

More in general, every straight line may be easily identified with the field \mathbb{K} . Indeed, a line R has equation of the form

$$ax + by + c = 0 \quad (7.11)$$

where a and b are not both equal to zero. Assume $b \neq 0$. Then we may *project* the line R on the x -axis, associating with each point (u, v) of R the point $(u, 0)$ of the x -axis, which will be identified with $u \in \mathbb{K}$ (see Figure 7.2). This

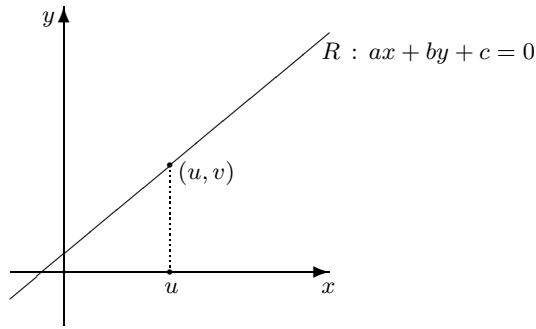


Fig. 7.2. Projection of a line R on the x -axis

mapping is bijective. Indeed, given u , we must have $v = -(au + c)/b$ if the point (u, v) is to lie on R . In other words, the projection is given by

$$\pi : (u, v) \in R \rightarrow u \in \mathbb{K}$$

and the inverse mapping is given by

$$\pi^{-1} : u \in \mathbb{K} \rightarrow (u, -(au + c)/b) \in R.$$

Analogously, if $a \neq 0$, the line R of equation (7.11) can be projected on the y -axis and the projection is bijective (see Exercise A7.14).

In conclusion, lines are not interesting from our viewpoint: in fact, recall that our goal, in cryptography, is to find groups *different* from \mathbb{K}^* .

The idea of projecting a curve on the x -axis to study it looks fine. So let us keep it. From this viewpoint, which are the simplest curves after the straight lines? We might answer, for instance, those for which the projection, even if not bijective, is *almost always* so, that is, is bijective but for a finite number of points. For instance, this property is enjoyed by the curves C of equation

$$g(x)y = f(x), \quad (7.12)$$

where $f(x)$, $g(x)$ are polynomials in x , with $f(x)$, $g(x)$ different from zero and without common factors. Every curve with these properties is irreducible (see Exercise A7.16). The projection is defined again as

$$\pi : (u, v) \in C \rightarrow u \in \mathbb{K}$$

and the *inverse mapping* is given by

$$\rho : u \in \mathbb{K} \rightarrow \left(u, \frac{f(u)}{g(u)}\right) \in R.$$

Notice that ρ is not defined where $g(u) = 0$, so, strictly speaking, it is not the inverse of π ; however it is its inverse out of finitely many points, the points $u \in \mathbb{K}$ such that $g(u) = 0$. Curves of this kind belong to the class of *rational curves*. These are irreducible curves C defined by an equation of the form (7.10), and such that there are rational functions $\phi(u)$, $\psi(u)$, defined over \mathbb{K} or an algebraic extension of \mathbb{K} , such that the rational function $f(\phi(u), \psi(u))$ is the zero function. In other words,

$$x = \phi(u), \quad y = \psi(u)$$

is a so-called *parametric representation* by rational functions of the curve C . For instance, the irreducible conic curves are curves of this kind (see Exercise A7.17).

Clearly, rational curves are again too similar to \mathbb{K} to be of interest to us, so we reject them too.

7.9.4 Hyperelliptic curves

The next case is given by curves for which the projection on the x -axis has no inverse, even after removing a finite number of points. Among these, the simplest case is that of curves for which the preimage of a point under the projection mapping consists in general not of a single point, but of two points. Curves of this kind are called *hyperelliptic*: examples of hyperelliptic curves are given by the irreducible curves C of degree greater than 2 having equation of the form

$$y^2 + yg(x) = f(x), \quad (7.13)$$

where $f(x), g(x) \in \mathbb{K}[x]$. Assume further that $g(x)$ is not the zero polynomial if the characteristic of \mathbb{K} is two: we shall shortly see why this hypothesis is necessary.

If (u, v) is a point on the curve C , this means that v is a solution of the equation

$$y^2 + yg(u) = f(u), \quad (7.14)$$

which has *in general* two distinct solutions.

Remark 7.9.2. Let us clarify the meaning of the previous claim.

If \mathbb{K} has characteristic different from 2, Equation (7.14) has a single solution if and only if its discriminant is zero, that is, if and only if

$$g(u)^2 + 4f(u) = 0. \quad (7.15)$$

It might happen that Equation (7.15) holds *for all* $(u, v) \in C$. However, if we suppose that the size of the set C' of points $u \in \mathbb{K}$ such that there is a point $(u, v) \in C$ is *large enough*, for instance that its size is greater than the degree of the polynomial $g(x)^2 + 4f(x)$, the factor theorem (see Theorem 1.3.19 and its Corollary 1.3.20) implies that $g(x)^2 + 4f(x)$ is the zero polynomial, and this yields a contradiction, because in this case we would find

$$y^2 + yg(x) - f(x) = \left(y + \frac{g(x)}{2}\right)^2,$$

against the hypothesis that the curve C is irreducible.

If the characteristic of \mathbb{K} is 2, the derivative of the polynomial $y^2 + yg(u) - f(u)$ is $g(u)$. Assume that *for all* $u \in C'$ we have $g(u) = 0$, so Equation (7.14) has a unique solution. Again, assuming that C' has size greater than the degree of $g(x)$, this would imply that $g(x)$ is the zero polynomial, contradicting the hypothesis.

Example 7.9.3. Consider the simple case in which $g(x)$ is the zero polynomial, which by hypothesis can happen only if the characteristic of \mathbb{K} is not 2. In this case, if (u, v) is a point on the curve C , this means that $v^2 = f(u)$, that is, $f(u)$ is a square in \mathbb{K} . So, not only (u, v) lies on C but $(u, -v)$ as well, and these are the only two points of C that project on the point $u \in \mathbb{K}$. They are *symmetric* with respect to the x -axis, in the sense that their second coordinates are one the opposite of the other. Of course, if $f(u) = 0$ these points coincide, otherwise they are distinct.

In conclusion, hyperelliptic curves can be thought of as *double coverings* of \mathbb{K} . This concept is particularly clear when \mathbb{K} is algebraically closed. In this case, by Remark 7.9.2, if \mathbb{K} has characteristic different from 2, for all $u \in \mathbb{K}$ that are not roots of the polynomial $g(x)^2 + 4f(x)$, we have exactly two points of C *over* the point $(u, 0)$ of the x -axis. If $u \in \mathbb{K}$ is a root of $g(x)^2 + 4f(x)$, the unique point $(u, -g(u)/2)$ of C corresponds to it.

If, on the other hand, \mathbb{K} has characteristic 2, for all $u \in \mathbb{K}$ that are not roots of the polynomial $g(x)$, we have exactly two points of C *over* the point $(u, 0)$ of the x -axis. If $u \in \mathbb{K}$ is a root of $g(x)$, the unique point $(u, \sqrt{f(u)})$ of C corresponds to it.

Clearly, we may consider curves for which the behaviour of the projection on the x -axis is even more complex: for instance, the preimage of a general point of \mathbb{K} may have size greater than two. But we shall not go into these cases because, as already remarked, we want to consider interesting curves which are nevertheless constructible in the easiest possible way.

7.9.5 Elliptic curves

So we are left with the problem of finding hyperelliptic curves which also are groups. This may be done, as we shall shortly see, if $f(x)$ and $g(x)$ have the simplest possible form compatible with the hypothesis that the curve has degree greater than 2. Indeed, assume that $g(x) = mx + n$ is of first degree and that $f(x) = x^3 + px^2 + qx + r$. In this case the curve C of equation (7.13) is cubic.

It is important to observe that, with suitable changes of variable, the equation of curve C may be simplified.

Proposition 7.9.4. *Let C be the curve of equation*

$$y^2 + y(mx + n) = x^3 + px^2 + qx + r. \quad (7.16)$$

It is possible to change coordinates in $\mathbb{A}_{\mathbb{K}}^2$ in such a way that in the new coordinate system

- *if \mathbb{K} has characteristic different from 2 or 3, C has equation of the form*

$$y^2 = x^3 + ax + b; \quad (7.17)$$

- *if \mathbb{K} has characteristic 3, C has equation of the form*

$$y^2 = x^3 + ax^2 + bx + c; \quad (7.18)$$

- *if \mathbb{K} has characteristic 2, C has equation of the form*

$$y^2 + cy = x^3 + ax + b; \quad (7.19)$$

or of the form

$$y^2 + xy = x^3 + ax^2 + b. \quad (7.20)$$

PROOF. To begin, assume \mathbb{K} not to have characteristic 2. Change variables as follows:

$$x \rightarrow x, \quad y \rightarrow \frac{y - mx - n}{2}.$$

The equation of C becomes of the form (7.16) where $m = n = 0$. This concludes the proof in the case of characteristic 3. If the characteristic is not 3, change again variables as follows:

$$x \rightarrow x - \frac{p}{3}, \quad y \rightarrow y. \quad (7.21)$$

The equation of C becomes now of the form (7.17), concluding the proof if the characteristic is neither 2 nor 3.

Assume now the characteristic of \mathbb{K} to be 2. By performing the change of variable (7.21) the equation of C becomes of the form

$$y^2 + y(mx + n) = x^3 + ax + b.$$

If $m = 0$ the equation is of the form (7.19). Assume then $m \neq 0$. In this case, perform the change of variable

$$x \rightarrow m^2x + \frac{n}{m}, \quad y \rightarrow m^3y + \frac{m^2a + n^2}{m^3},$$

obtaining an equation of the form (7.20). \square

The equations of the form (7.17), (7.18), (7.19), (7.20) are called canonical equations in *Weierstrass form* of a cubic curve.

Now we shall put ourselves in a *regularity hypothesis*. We shall assume that, if \mathbb{K} has characteristic different from 2, the right-hand side of Equation (7.17) or (7.18) has no multiple roots in the algebraic closure of \mathbb{K} . If \mathbb{K} has neither characteristic 2 nor 3, that is, when the equation in Weierstrass form is Equation (7.17), this is equivalent to saying that $27b^2 + 4a^3 \neq 0$ (see Exercise A7.18). If, on the other hand, \mathbb{K} has characteristic 2 and if the equation in Weierstrass form is Equation (7.20), then we shall assume $b \neq 0$. We shall shortly see the meaning of this hypothesis.

Finally, we shall add to C a point O called *point at infinity*, whose meaning is well known to the reader acquainted with projective geometry (see [51]). Next, we shall denote by E the set $C \cup \{O\}$, call E an *elliptic curve*, and say that (7.17), (7.18), (7.19) or (7.20) is its equation.

Remark 7.9.5. As is well known, the affine plane $\mathbb{A}_{\mathbb{K}}^2$ can be naturally embedded in the *projective plane* $\mathbb{P}_{\mathbb{K}}^2$, whose points are non-zero ordered triples $[x_0, x_1, x_2]$ of elements of \mathbb{K} , up to a multiplication by a constant, that is, $[x_0, x_1, x_2] = [kx_0, kx_1, kx_2]$ for all $k \in \mathbb{K}^*$. Given the point $[x_0, x_1, x_2]$ of $\mathbb{P}_{\mathbb{K}}^2$, x_0, x_1, x_2 are said to form a triple of *homogeneous coordinates* of the point. The embedding of $\mathbb{A}_{\mathbb{K}}^2$ in $\mathbb{P}_{\mathbb{K}}^2$ happens as follows:

$$(x, y) \in \mathbb{A}_{\mathbb{K}}^2 \rightarrow [1, x, y] \in \mathbb{P}_{\mathbb{K}}^2.$$

The complement of $\mathbb{A}_{\mathbb{K}}^2$ in $\mathbb{P}_{\mathbb{K}}^2$ is the set of points satisfying the *equation* $x_0 = 0$, that is, the set of points of the form $[0, a, b]$, called *points at infinity*. This set is called *line at infinity* of the projective plane. If $[x_0, x_1, x_2]$ is not on the line at infinity, its cartesian coordinates in $\mathbb{A}_{\mathbb{K}}^2$ are

$$x = \frac{x_1}{x_0}, \quad y = \frac{x_2}{x_0}.$$

These are the formulas to pass from homogeneous coordinates to cartesian coordinates.

If we consider a line R in the plane $\mathbb{A}_{\mathbb{K}}^2$, with equation $bx - ay + k = 0$, passing to homogeneous coordinates and multiplying both sides by x_0 , we find the equation $kx_0 + bx_1 - ax_2 = 0$. Clearly all the solutions to this equation having $x_0 \neq 0$, and they alone, correspond to the points of $\mathbb{A}_{\mathbb{K}}^2$ lying on R . On the other hand, by intersecting it with the line at infinity, we obtain the system

$$x_0 = bx_1 - ax_2 = 0,$$

which uniquely determines the point $0 = [0, a, b]$. This leads to the well-known interpretation of the points at infinity: the point $[0, a, b]$ is to be considered as *the common point* of all parallel lines of the plane $\mathbb{A}_{\mathbb{K}}^2$ having equation of the form $bx - ay + k = 0$ with k ranging in \mathbb{K} .

Similarly, considering the curve C of equation (7.17), passing to homogeneous coordinates and multiplying both sides by x_0^3 , we find the equation

$$x_0 x_2^2 = x_1^3 + a x_0^2 x_1 + b x_0^3. \quad (7.22)$$

All the solutions to this equation having $x_0 \neq 0$, and they alone, correspond to the points of $\mathbb{A}_{\mathbb{K}}^2$ lying on C . On the other hand, intersecting it with the line at infinity, we obtain the system

$$x_0 = x_1 = 0,$$

which uniquely determines the point $0 = [0, 0, 1]$. So it is natural to consider Equation (7.22) as defining the *projective closure* E of C . It differs from C only by the point at infinity O . We may reason analogously if the curve has equation (7.18), (7.19) or (7.20).

Remark 7.9.6. We might wonder whether a curve having equation (7.17), (7.18), (7.19) or (7.20) could itself be rational, and so devoid of interest for our uses. It is not so in the regularity hypothesis we have stipulated: for instance when \mathbb{K} has characteristic different from 2 and 3, the equation is of the form (7.17) and $27b^2 + 4a^3 \neq 0$, while it can be shown that the curve is rational if $27b^2 + 4a^3 = 0$ (see Exercise A7.23). The simplest case is that of the curve of equation $y^2 = x^3$, which has parametric representation

$$x = u^2, \quad y = u^3.$$

To study the matter more in depth, see [56].

7.9.6 Group law on elliptic curves

Let us discuss now the group law on an elliptic curve E . We shall consider here in detail the case in which \mathbb{K} has characteristic different from 2 or 3 and the equation is of the form (7.17), leaving to the reader as an exercise the analogous discussion of the remaining cases (see Exercises A7.27 and [56], Chapter III, § 2).

The key observation is that *given two points $p = (x_1, y_1)$ and $q = (x_2, y_2)$ of the curve, the line through them intersects the curve in a third point $r = (x_3, y_3)$* . This observation is to be taken with a grain of salt, in the sense we are going to explain.

First of all, we verify it in the case in which p and q are distinct and the line R through them is not *vertical*, that is, has not an equation of the form $x = u$. This means, as already remarked, that $x_1 \neq x_2$.

The equation of R is

$$y = mx + n, \quad (7.23)$$

with

$$m = \frac{y_2 - y_1}{x_2 - x_1}, \quad n = y_1 - mx_1 \quad (7.24)$$

(see Exercise A7.19). To find the points of intersection of R with C , substitute (7.23) in (7.17), and solve with respect to x . So one gets the third degree equation

$$x^3 - (mx + n)^2 + ax + b = 0;$$

clearly x_1 and x_2 are two of its roots. Let x_3 be the third root. As

$$x_1 + x_2 + x_3 = m^2$$

(see Exercise A7.20), we have

$$x_3 = m^2 - x_1 - x_2,$$

which gives the first coordinate of the third point r of intersection of R with C . The second coordinate of r is given by

$$y_3 = mx_3 + n.$$

Let us see what happens if the line through two distinct points p and q is the vertical line $x = u$. As we have seen in Example 7.9.3, this means that the two points have coordinates (u, v) and $(u, -v)$, where $\pm v$ are the square roots of $u^3 + au + b$. The intersection of the line of equation $x = u$ with the curve of equation (7.17) in $\mathbb{A}_{\mathbb{K}}^2$ consists only of the points p and q . But passing to homogeneous coordinates, the equation of the line becomes $x_1 = ux_0$ and we see that it passes through the point O lying in E . So it is natural to regard O as the third intersection point of the line through p and q with the curve.

Example 7.9.7. We demonstrate the preceding remarks by examining the real curve of equation

$$y^2 = x^3 - x.$$

The curve corresponds to the union of the graph of the function

$$y = \sqrt{x^3 - x}$$

and of its symmetric with respect to the x -axis. Fix a point on the curve, say $p = (2, \sqrt{6})$. Write the equation of a non-vertical line through p . It is of the form

$$y - \sqrt{6} = \frac{1}{m}(x - 2), \quad (7.25)$$

with $m \neq 0$. Intersect this line with the curve, obtaining, besides p , two more points, q and r . We leave to the reader the task of finding their coordinates as functions of m and of verifying that, as m approaches 0, that is, when the line tends to becoming the vertical line $x = 2$, one of the two points q and r tends to the point $p' = (2, -\sqrt{6})$, symmetric of p with respect to the x -axis, which lies indeed on the vertical line $x = 2$, while the other's second coordinate tends to infinity (see Figure 7.3). Basically, this second point tends to infinity and its *limit position*, which may be thought of as infinitely far along the y -axis, to which the line of equation (7.25) becomes parallel as m approaches 0, is exactly that of the point O we have added to C in order to get E . These heuristic remarks are quite natural and should not sound strange to readers acquainted with projective geometry.

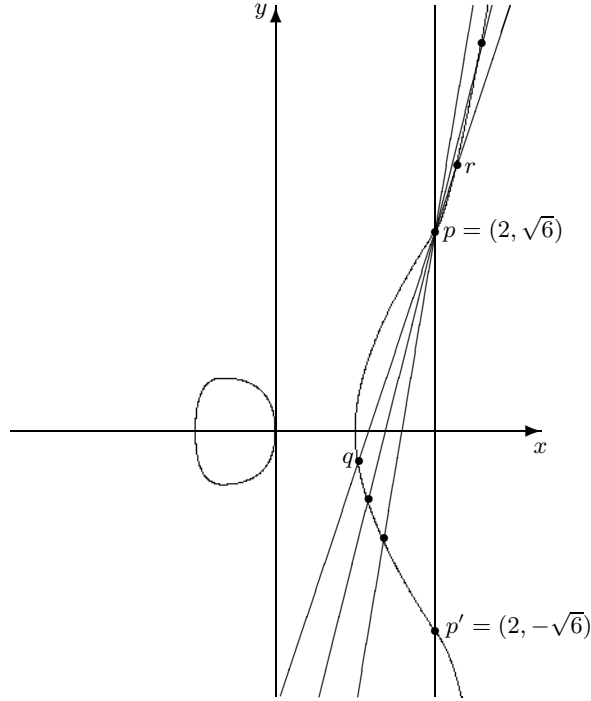


Fig. 7.3. Elliptic curve of equation $y^2 = x^3 - x$

Finally, what happens if $p = q$? Here we cannot consider the *line through* p and q . Nevertheless, among the infinitely many lines through $p = q$ one is special, with respect to the elliptic curve: the *tangent line* to the curve in p , which may be thought of as the line joining p with a point q on the curve that is *so close to p to be undistinguishable from p* . This notion is well known in the real case: it is the limit line of the line through p and another point q of the curve, when q approaches p .

If a real curve C has equation

$$f(x, y) = 0$$

and if $p = (\xi, \eta)$ is a point of C , the condition for the tangent, seen as the above limit, to exist is that in (ξ, η) not both partial derivatives of $f(x, y)$ are zero (see [51]), that is, it is not the case that

$$\frac{\partial f}{\partial x}(\xi, \eta) = 0, \quad \frac{\partial f}{\partial y}(\xi, \eta) = 0. \quad (7.26)$$

So the tangent line to C in p has equation

$$\frac{\partial f}{\partial x}(\xi, \eta)(x - \xi) + \frac{\partial f}{\partial y}(\xi, \eta)(y - \eta) = 0. \quad (7.27)$$

More in general, these notions extend without any difference to the case of a curve on an arbitrary field (see [56], Ch. I, § 1). A point $p = (\xi, \eta)$ of the curve C of equation $f(x, y) = 0$ for which (7.26) hold is said to be *singular*. In it the tangent line does not exist. A non-singular point is said to be *simple* or *smooth*, and in it the tangent line exists and is given by Equation (7.27). A curve having a singular point in p is said to be *singular* in p .

Remark 7.9.8. A curve of equation $f(x, y) = 0$ is *singular*, that is, has some singular point, if and only if the system

$$f(x, y) = 0, \quad \frac{\partial f}{\partial x}(x, y) = 0, \quad \frac{\partial f}{\partial y}(x, y) = 0 \quad (7.28)$$

admits solutions. Notice that this definition depends on the field \mathbb{K} , as it is possible that the system (7.28) has no solutions in \mathbb{K} but has solutions in some extension of \mathbb{K} (see Exercise B7.58).

When a curve is said to be *non-singular*, without specifying the field on which it is considered, the curve is meant to be considered on the algebraic closure of the field \mathbb{K} containing the coefficients of the equation defining the curve.

The reader may easily verify that the regularity hypothesis on page 373 makes sure that the curves C defined by equations in Weierstrass form of the kind (7.17), (7.18), (7.19) or (7.20) are non-singular (see Exercise A7.22). Let us fix our attention, as usual, on the case in which the equation is of the form (7.17). Then, given a point $p = (\xi, \eta)$ of the curve, the tangent line R_p in that point has equation

$$(-3\xi^2 - a)(x - \xi) + 2\eta(y - \eta) = 0.$$

It is vertical if and only if $\eta = 0$, that is on the points in which C intersects the x -axis. We leave to the reader the task of verifying that if $\eta \neq 0$, then R_p intersects C in a further point r the coordinates of which may be easily computed (see Exercise A7.25). The point r is to be interpreted as the third intersection with the elliptic curve of the *line through* p and q when $p = q = (\xi, \eta)$. Of course, if $p = q = (\xi, 0)$, the tangent line is the vertical line of equation $x = \xi$ and, as we have already seen, it is the point at infinity O that has to be regarded as the third intersection of this line with the curve.

What can be said about the lines through O ? By our interpretation of points at infinity, they are all the lines that are parallel to the y -axis, that is, the vertical lines of equation $x = u$.

So, if $p = (\xi, \eta)$ lies on the curve, the line through p and O is the line of equation $x = \xi$, which, as we know, intersects the curve in the further point $q = (\xi, -\eta)$.

Finally, for reasons we shall not discuss at length (see Exercise A7.26), it can be seen that the line at infinity must be considered as the tangent line to the curve E in O and, as already seen, it intersects the curve only in O . So we

may say that this line, which must be considered as the line through p and q when $p = q = O$, intersects further the curve in O itself.

In conclusion, we may say that, in the sense made clear above, given two points p and q of E , distinct or equal, there is a third point r of E such that p , q and r are *collinear*.

So we are very close to defining the group law on E , which will be described using an additive notation. One is tempted to define the *sum* $p + q$ of two points p and q of E as the third point of E that is collinear with p and q . But this is not a completely correct idea. We have to take O as the identity element, that is the *zero*, of the group, and define the sum $p + q$ as follows:

- consider first the *third point* r of E collinear with p and q ;
- define $p + q$ as the *third point* s of E collinear with r and O .

In other words, if we consider an elliptic curve defined by an equation of the form (7.17), $p + q$ is the symmetric point with respect to the x -axis of the third point r of E collinear with p and q (see Figure 7.4). In this way, the opposite of each point p is exactly its symmetric with respect to the x -axis.

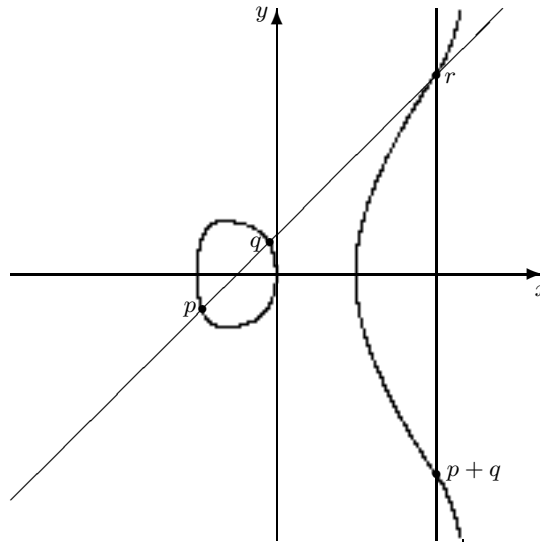


Fig. 7.4. Group law on an elliptic curve

Keeping in mind what has been said in this section, as well as in Exercise A7.25, we may compute the coordinates of the point $s = p + q = (x_3, y_3)$ as a function of the points $p = (x_1, y_1)$ and $q = (x_2, y_2)$ of an elliptic curve (see Exercise A7.27). Here we give the details only for the case in which the equation in Weierstrass form is of the kind (7.17).

Proposition 7.9.9. *Let E be an elliptic curve on a field \mathbb{K} of characteristic different from 2 and 3, of equation $y^2 = x^3 + ax + b$ with $27b^2 + 4a^3 \neq 0$. Consider the binary operation $+$ in E*

$$E \times E \longrightarrow E, \quad (p, q) \longrightarrow s$$

defined as follows:

- if $p = O$, then $s = q$;
- if $p \neq q$ are both different from O , $p = (x_1, y_1)$ and $q = (x_2, y_2)$, and if $x_1 = x_2$ and $y_1 = -y_2$, then $s = O$;
- if $p \neq q$ are both different from O , $p = (x_1, y_1)$ and $q = (x_2, y_2)$ with $x_1 \neq x_2$, then $s \neq O$ and $s = (x_3, y_3)$ with

$$x_3 = \left(\frac{y_2 - y_1}{x_2 - x_1} \right)^2 - x_1 - x_2, \quad y_3 = \frac{(y_2 - y_1)}{(x_2 - x_1)}(x_1 - x_3) - y_1;$$

- if $p = q$ is different from O and $p = (\xi, 0)$, then $s = O$;
- if $p = q$ is different from O and $p = (\xi, \eta)$, with $\eta \neq 0$, then $s \neq O$ and $s = (\xi', \eta')$ with

$$\xi' = \left(\frac{3\xi^2 + a}{2\eta} \right)^2 - 2\xi, \quad \eta' = \frac{(3\xi^2 + a)}{2\eta}(\xi - \xi') - \eta.$$

With this operation, $(E, +)$ is an abelian group, whose identity element is the point at infinity O and the opposite of the point (ξ, η) of E is the point $(\xi, -\eta)$.

PROOF. The commutativity of the operation just defined is a simple algebraic computation we leave to the reader as an exercise (see Exercise A7.28). The most delicate check is that the operation is associative: for it see [56], Proposition 2.2. \square

Remark 7.9.10. Notice that without the hypothesis $27b^2 + 4a^3 \neq 0$, we may have anomalous situations, as the following one. Let E be the real curve of equation

$$y^2 = x^3 - 3x - 2.$$

We have $27 \cdot 4 + 4 \cdot (-3)^3 = 0$. Now it is easy to verify that, if we keep the definition of the sum as given above, for every $(x, y) \in E$ we have

$$(-1, 0) + (x, y) = (-1, 0).$$

7.9.7 Elliptic curves over \mathbb{R} , \mathbb{C} and \mathbb{Q}

We have finally come in possession of a *group for each elliptic curve*, and we may use these groups in cryptography, as we intended. But which ones of these curves should we use? and over which fields?

Real elliptic curves possess *infinitely many points*. Indeed, a real elliptic curve E is defined by the equation (7.16). The curve E consists of the graph of the function $y = \sqrt{f(x)}$ and of its symmetric with respect to the x -axis. As x approaches $+\infty$, $f(x)$ tends to $+\infty$. Thus $\sqrt{f(x)}$ is defined at least on a half-line, so E has infinitely many points. It is easily verified that E consists of either one or two arcs, depending on the number (one or three) of real roots of $f(x)$ (see Exercise A7.29). In cryptography we need finite groups, so real elliptic curves are useful to draw inspiration from, but cannot be used for our goals.

Elliptic curves over \mathbb{C} are even less useful. \mathbb{C} being algebraically closed, those curves are, as mentioned, *double covers* of \mathbb{C} . They can be parametrised using suitable functions, called *elliptic functions*, which are not rational functions, but have properties quite similar to those of trigonometric functions and cannot be expressed in terms of elementary functions. The theory of these functions is very interesting but too complex to allow us more than the briefest mention (see [55]). It is interesting to remark that elliptic curves take their name from these functions.

We may next consider elliptic curves over \mathbb{Q} . In this regard, the following fundamental theorem is well known (see [56], pag. 188):

Theorem 7.9.11 (Mordell–Weil). *If E is an elliptic curve over \mathbb{Q} , then E is a finitely generated abelian group, that is,*

$$E \simeq \text{Tors}(E) \oplus \mathbb{Z}^r,$$

where $\text{Tors}(E)$ is the torsion subgroup of E , that is to say, the subgroup of E consisting of the points of finite order, while r is called rank of E .

The dependence of the rank of an elliptic curve over \mathbb{Q} from its equation is not yet well understood.

Example 7.9.12. The point $p = (2, 3)$ of the elliptic curve over \mathbb{Q} of equation $y^2 = x^3 + 1$ is a torsion point. Indeed, by using the group law on the curve we find that $2p = p + p = (0, 1)$, $4p = (0, -1)$ and so $6p = O$.

Excluding the case in which the rank of a rational elliptic curve E is 0, E is an infinite group too, and so unsuitable for use in cryptography.

So we are only left with elliptic curves on finite fields \mathbb{F}_q . This is a classical and intriguing subject which has played a central role in last century's mathematics, culminating in the momentous proof by A. Wiles of the well-known so-called Fermat's Last Theorem, which states that the equation $x^n + y^n = z^n$ has no solutions $(x, y, z) \in \mathbb{Z} \times \mathbb{Z} \times \mathbb{Z}$ with x, y, z different from zero, if $n > 2$ (see the popular science book [57] or Wiles's paper [63]; see also Exercises A7.30–A7.35 for the case $n = 2$).

7.9.8 Elliptic curves over finite fields

On the road to describe elliptic curves over a finite field \mathbb{F}_q , we begin by remarking that such a curve E , of equation (7.16), has finitely many points, their number being denoted by $|E/\mathbb{F}_q|$. We also have an estimate

$$|E/\mathbb{F}_q| \leq 2q + 1 \quad (7.29)$$

because, apart from O , for all $x \in \mathbb{F}_q$, the equation (7.16) has at most two solutions in \mathbb{F}_q .

The estimate (7.29) is very rough: indeed, only one half of the elements of \mathbb{F}_q are squares, so only one half of the elements of \mathbb{F}_q has a square root. To be more precise, we can give the following definition, which is the analogous, for finite fields, of the Legendre symbol:

Definition 7.9.13. *The quadratic character in \mathbb{F}_q is the function*

$$\chi : u \in \mathbb{F}_q \rightarrow \chi(u) \in \{0, 1, -1\}$$

defined as follows:

$$\chi(u) = \begin{cases} 0 & \text{if } u = 0, \\ 1 & \text{if } u \text{ is a square,} \\ -1 & \text{if } u \text{ is not a square.} \end{cases}$$

In particular, $\chi(u) = (\frac{u}{q})$ if q is a prime number.

Notice that the number of solutions of the equation $x^2 = u$ in \mathbb{F}_q equals $1 + \chi(u)$ (see Exercise A7.36). Moreover, $\chi(uv) = \chi(u)\chi(v)$ for every pair (u, v) of elements of \mathbb{F}_q (see Exercise A7.37).

Then, if the characteristic of \mathbb{F}_q is different from 2 and 3 (hence the equation becomes (7.17)), we have

$$|E/\mathbb{F}_q| = 1 + \sum_{x \in \mathbb{F}_q} (1 + \chi(x^3 + ax + b)) = q + 1 + \sum_{x \in \mathbb{F}_q} \chi(x^3 + ax + b).$$

Now, let us reason heuristically: we expect that, for general a and b , for a given $x \in \mathbb{F}_q$, $\chi(x^3 + ax + b)$ has the same probability of being equal to 1 or -1 . That is, for all $x \in \mathbb{F}_q$, computing $\chi(x^3 + ax + b)$ is like tossing a coin to see whether it shows heads or tails.

In probability theory a situation of this kind is called *random walk*. Assume we are on a line, in the coordinate origin. We have a coin and we toss it. If it shows heads, we take a step towards the positive semiaxis, while if it shows tails we take a step towards the negative semiaxis. After n steps, how far may we expect to be from the origin? The answer given by probability theory is that we expect a distance of about \sqrt{n} steps, in one of the two directions (see [29]).

In our case $n = q$, that is, the number of points of \mathbb{F}_q . So we expect, by this heuristic argument, that the number $|E/\mathbb{F}_q|$ of points in the elliptic curve E is *on the average* bounded by $q + 1 + \sqrt{q}$. The following result by Hasse (see [56], p. 131) gives us an actual, not just a heuristic, estimate for $|E/\mathbb{F}_q|$. Notice that this estimate is not very far from the previous one.

Theorem 7.9.14 (Hasse's Theorem). *If $N = |E/\mathbb{F}_q|$, then*

$$|N - (q + 1)| \leq 2\sqrt{q}.$$

Hasse's theorem says that an elliptic curve over \mathbb{F}_q has, after all, *not many more points* than \mathbb{F}_q itself. So elliptic curves over finite fields are actually *not too complicated objects*. Good news for cryptographers!

Example 7.9.15. Let us compute now the number of points of the elliptic curve of equation $y^2 = x^3 + x$ over \mathbb{F}_p , with p a prime number such that $p \equiv 3 \pmod{4}$. We have

$$N = p + 1 + \sum_{x \in \mathbb{F}_p} \chi(x^3 + x) = p + 1 + \sum_{x \in \mathbb{F}_p^*} \chi(x^3 + x).$$

But

$$\begin{aligned} \chi((-x^3) + (-x)) &= \chi((-1)(x^3 + x)) = \chi(-1)\chi(x^3 + x) = \\ &= \left(\frac{-1}{p}\right) \chi(x^3 + x) = -\chi(x^3 + x); \end{aligned}$$

hence it follows that $N = p + 1$, as the summands in the sum giving N cancel out in pairs.

A result more precise than Hasse's Theorem is Weil's theorem. It is one of the most important theorems of 20th century mathematics; it led Weil to conjecture more general results which were later proved by Deligne, giving a great boost both to algebraic geometry and number theory.

To state Weil's theorem (see [56], Ch. V, §2), associate with an elliptic curve E defined over \mathbb{F}_q a function called *zeta function* of E , denoted by $Z_{E, \mathbb{F}_q}(t)$. If $N_r = |E/\mathbb{F}_{q^r}|$, define

$$Z_{E, \mathbb{F}_q}(t) := e^{\sum_{r=1}^{\infty} N_r t^r / r}.$$

Theorem 7.9.16 (A. Weil). *The function $Z_{E, \mathbb{F}_q}(t)$ is rational, of the form*

$$\frac{1 - at + qt^2}{(1 - t)(1 - qt)} \quad (7.30)$$

and only depends on E . More precisely,

$$a = q + 1 - N_1.$$

Notice that the discriminant $\Delta = a^2 - 4q$ of the numerator of (7.30) is negative by Hasse's theorem, so the latter has two complex conjugate roots α', β' . It is easy to verify, and we leave it as an exercise to the reader (see Exercise A7.38), that, by setting

$$\alpha = \frac{1}{\alpha'}, \quad \beta = \frac{1}{\beta'},$$

we have $|\alpha| = |\beta| = \sqrt{q}$.

Corollary 7.9.17. *For all r we have*

$$N_r = 1 + q^r - \alpha^r - \beta^r.$$

PROOF. From Weil's theorem it follows that:

$$\begin{aligned} \sum_{r=1}^{\infty} N_r \frac{t^r}{r} &= \log \frac{(1 - \alpha t)(1 - \beta t)}{(1 - t)(1 - qt)} = \\ &= \log(1 - \alpha t) + \log(1 - \beta t) - \log(1 - t) - \log(1 - qt). \end{aligned}$$

Taking derivatives on both sides we find

$$\begin{aligned} \sum_{r=1}^{\infty} N_r t^{r-1} &= -\frac{\alpha}{1 - \alpha t} - \frac{\beta}{1 - \beta t} + \frac{1}{1 - t} + \frac{q}{1 - qt} = \\ &= -\alpha \sum_{r=0}^{\infty} (\alpha t)^r - \beta \sum_{r=0}^{\infty} (\beta t)^r + \sum_{r=0}^{\infty} t^r + q \sum_{r=0}^{\infty} (qt)^r = \\ &= \sum_{r=0}^{\infty} (-\alpha^{r+1} - \beta^{r+1} + 1 + q^{r+1}) t^r, \end{aligned}$$

hence the corollary immediately follows. \square

Let us see how to apply these results to computing the number of points on an elliptic curve on a finite field.

Example 7.9.18. Let us compute the number N_1 of points over \mathbb{F}_2 , and the number N_r of points over any finite extension of degree r , of the elliptic curve E of equation $y^2 + y = x^3 + 1$. Computing the number of points of any curve over \mathbb{F}_2 is trivial. We may easily proceed by trial and error, keeping in mind that in the affine plane there are exactly 4 points. Then Weil's theorem gives the number of points of the curve over any extension of \mathbb{F}_2 .

In this case we have $N_1 = 3$, because E , apart from the point at infinity O , has over \mathbb{Z}_2 only the points $(1, 0)$ and $(1, 1)$ (see Exercise B7.66). Then the zeta function is

$$Z(t) = \frac{1 + 2t^2}{(1 - t)(1 - 2t)}.$$

The roots of the numerator are $\pm i/\sqrt{2}$, so

$$N_r = 1 + 2^r - (i\sqrt{2})^r - (-i\sqrt{2})^r = \begin{cases} 1 + 2^r & \text{if } r \text{ is odd,} \\ 1 + 2^r - 2(-2)^{r/2} & \text{if } r \text{ is even.} \end{cases}$$

7.9.9 Elliptic curves and cryptography

Let us come back to cryptography. Now we have at our disposal not only the multiplicative groups of the fields \mathbb{F}_q , but also the elliptic curves defined on them, and there are many more of them. So, as already mentioned, we have more diversity, and this gives more security to our cryptosystems.

But are elliptic curves on finite fields actually appropriate for cryptography? The answer is essentially in the affirmative. Let us see why, discussing separately the issues already hinted at at the beginning of this section.

- Elliptic curves are given in a concrete way: it suffices to give their equation. This does not always mean that determining their points is easy. Indeed, there is no known polynomial algorithm to generate points on an elliptic curve over \mathbb{F}_q . However, there are *probabilistic* polynomial algorithms, which are able to determine points on an elliptic curve with a very high probability (see Exercise A7.39 and A7.40). Let us further mention that there is in fact a polynomial algorithm, due to R. Schoof, that determines the *number* of points of an elliptic curve over \mathbb{F}_q , but without determining the points themselves (see [50]).
- Exponentiating on an elliptic curve E , which amounts now actually to multiplying a point of E by an integer, has a polynomial computational cost: keeping in mind Proposition 7.9.9, this can be shown in a similar way as in \mathbb{Z}_p or in \mathbb{F}_q .
- A. Menezes, T. Okamoto and S. A. Vanstone ([42]) have shown that the problem of discrete logarithms on an elliptic curve is not less hard than on a finite field. It is conjectured that, a fortiori, for an elliptic curve on \mathbb{F}_q the Diffie–Hellman hypothesis holds.

In conclusion, let us sum up how it is possible to exchange a private key using an *RSA* system relying on the use of an elliptic curve. We choose: a field \mathbb{F}_q , an elliptic curve E and a point $p \in E$, which are made public. As we shall see shortly, it is convenient for the system to work best, that E has many points over \mathbb{F}_q . To determine such an E , the results described above are helpful.

Each user U chooses a key e_U , a positive integer, he will keep private. However he publishes $p_U = e_U p$, which is again a point of E , computed by U in polynomial time. It is convenient that if $U \neq V$ then $p_U \neq p_V$. To this end, it helps if p has a very large order, far larger than the number of users of the system. So, when the numbers e_U are chosen randomly the points p_U will be different.

If two users A and B want to exchange a private key, they may do so by agreeing about using as their private key the point $p_{AB} = (e_A e_B)p$. They both may determine it in polynomial time, while, due to the difficulty of computing discrete logarithms on E and Diffie–Hellman hypothesis, p_{AB} will be unreachable by anybody else.

Clearly, A and B may use as their private key a number deduced from p_{AB} : for instance, one of the coordinates of this point, or their sum, and so on.

We leave to the reader the task of devising a way of applying the general *RSA* method using elliptic curves (see Exercise A7.41).

7.9.10 Pollard's $p - 1$ factorisation method

Surprisingly, elliptic curves are not only greatly useful in cryptography, but are also suitable to solving several other problems we have previously discussed, such as primality tests (S. Goldwasser, J. Kilian; A.O.L. Atkin, see [30], Ch. VI) and factorisation (H.W. Lenstra, see [30], l.c.). The ideas are not very different from those described in this book, but elliptic curves make it possible to implement them with an accuracy and a flexibility that make them very effective. We conclude this part by examining a factorisation method (Pollard's $p - 1$ method) which does not rely on elliptic curves. Without going into too many details, we mention the fact that the restrictions of this method can be overcome by using elliptic curves, and this yields Lenstra's factorisation mentioned.

Assume we have to factor an integer N . The method we are going to describe works when N has a prime factor p , to be determined, such that $p - 1$ has not too large prime divisors. So we give an a priori estimate of the greatest prime T dividing $p - 1$.

We have to find a number k divided by $p - 1$. To this end, we may proceed as follows. Let

$$p - 1 = 2^\alpha 3^\beta 5^\gamma \dots T^\omega$$

be the prime factorisation of $p - 1$. As p is not known, the exponents α, β, γ etc. are not known either. However, as $2^\alpha \leq p - 1 < N$, we have $\alpha < (\log N)/(\log 2)$, and so $\beta < (\log N)/(\log 3)$, etc. Thus, setting

$$k := 2^{\lfloor (\log N)/(\log 2) \rfloor} \cdot 3^{\lfloor (\log N)/(\log 3) \rfloor} \dots T^{\lfloor (\log N)/(\log T) \rfloor},$$

we have that $p - 1$ divides k .

Let now a be an integer between 2 and $N - 2$, such that $\text{GCD}(a, N) = 1$, so a is relatively prime with all prime factors of N . By Fermat's little theorem we have $a^{p-1} \equiv 1 \pmod{p}$ and so, as $p - 1 | k$,

$$a^k \equiv 1 \pmod{p}.$$

Compute now $a^k \pmod{N}$ in polynomial time (see § 3.3.1) and simultaneously, using the Euclidean algorithm, compute $d = \text{GCD}(a^k - 1, N)$. Clearly, the first p we are looking for is such that $p | d$, so $d \neq 1$. If $d \neq N$, we have found a (not necessarily prime) factor of N and we are done. Otherwise, if $d = N$, that is, if $N | (a^k - 1)$, modify the choice of the integer a or of the integer k and start over. In practice, take as k a multiple of all integers smaller than or equal to a fixed integer M , which is supposed to be greater than all the powers of the prime numbers dividing $p - 1$. For instance, we may take $k = M!$.

Example 7.9.19. Factor with this method the number 156203. Choose $M = 6$, $k = 6!$, and $a = 2$. Compute $2^k \pmod{156203} = 32219$. Find next

$$\text{GCD}(2^k - 1, 156203) = 181.$$

We have found that 181 (which is a prime) is a factor of 156203 and the factorisation of 156203 is

$$156203 = 181 \cdot 863.$$

Remark 7.9.20. While the single steps just described have polynomial cost, the algorithm itself is exponential because, when $d = N$, we have to modify, for instance, our choice of the integer a , which may be done in N ways. Moreover, it may happen that for each choice of a we have $N \mid (a^k - 1)$, and so the algorithm might never give a positive result.

Nevertheless, there are probabilistic reasons for this algorithm to work in some cases. For instance, suppose exactly one of the prime factors of N , say p , has the property that the prime factors of $p - 1$ are bounded by T , while for all other factors q , q is large with respect to k .

In this situation, if $q \mid (a^k - 1)$ then a is a k th root of unity modulo q , and the probability of this happening for a random choice of a is k/q , because the k th roots of unity are at most k in \mathbb{Z}_q^* . So this probability is very small, if $q \gg k$. Hence the probability that $N \mid (a^k - 1)$ is even smaller, and this is exactly the case in which the algorithm has to be repeated.

Which are the limitations of this algorithm? In it, we exploit the structure of the groups \mathbb{Z}_p^* , with p ranging among the prime factors of N . For a fixed N , these groups are fixed and cannot be exchanged for others; and, as we have remarked, the algorithm might not give positive result for any of them. This happens, in particular, if the order $p - 1$ of each of these groups has at least a prime factor not bounded by the number T we have chosen at the beginning and which, as seen, determines the number k . How may we obtain a larger choice? By using elliptic curves, on which H. W. Lenstra's factorisation method relies (see [35], [30], Ch. VI, § 4). Here is a sketch of the idea: substitute the group E/\mathbb{Z}_p of the points of an elliptic curve E over \mathbb{Z}_p for \mathbb{Z}_p^* . The new group, by Hasse's theorem 7.9.14, has order

$$|E/\mathbb{Z}_p| = p + 1 - s, \quad \text{with } |s| \leq 2\sqrt{p}.$$

Different elliptic curves E yield different values of s and we have at our disposal several groups: so it is realistic to expect one of them to have order with small prime factors.

Appendix to Chapter 7

A7 Theoretical exercises

A7.1. We have seen the frequencies of the different letters in English. Assume we have messages written in two languages, for instance containing an English text and its translation in another language, or the other way around. Explain how to get a frequency table for these messages, assuming we have the tables for the two languages, knowing that each message is written half in one of the languages and half in the other one.

A7.2. We have seen how to carry out the cryptanalysis of an enciphered message by using frequency analysis. Exercise C7.2 requests the reader to write a program that computes the frequencies with which the letters appear in a given text. Assume we have a program saying whether a word exists in English language or not: more precisely, on receiving as its input a word, its output is either *true* or *false*, depending on the word being present or not in English lexicon. Describe an algorithm to decrypt a message, enciphered using a monoalphabetic substitution, using these two programs.

A7.3. Prove that an affine transformation $f : \mathbb{Z}_{26} \rightarrow \mathbb{Z}_{26}$ defined by $f(n) = an + b \pmod{26}$, where $a, b \in \mathbb{Z}$, is bijective if and only if $\text{GCD}(a, 26) = 1$.

A7.4.* Show by an example that there are systems of two linear congruences modulo 26 in two unknowns having no solutions. (Hint: try a diagonal matrix. Indeed, in this case the system reduces to two independent linear congruences; the system does not admit solutions if and only if one of the two congruences does not.)

A7.5.* Show by an example that there are systems of two linear congruences modulo 26 in two unknowns having more than one solution modulo 26. (Hint: try a diagonal matrix.)

A7.6. Explain how to carry out the cryptanalysis of an affine cipher, with 2-letter unitary messages, when the system of linear congruences modulo 26 obtained as explained in Remark 7.4.3 on page 339 has more than one solution.

A7.7.* Prove Proposition 7.4.2 on page 338.

A7.8.* Consider a system of two linear congruences of the form

$$A \cdot \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix} \equiv \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{pmatrix} \pmod{m},$$

where A is a square $n \times n$ matrix with integer coefficients. Prove that if the determinant of A is invertible modulo m , then A has an inverse modulo m and the system admits a unique solution in a_1, a_2, \dots, a_n , given by

$$\begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix} \equiv A^{-1} \cdot \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{pmatrix} \pmod{m}.$$

A7.9. Prove that the computational complexity of the knapsack problem, when a_1, \dots, a_n is a superincreasing sequence of integers, is polynomial.

A7.10.* Assume we know that an integer n is the product of two prime numbers p and q . Explain how to find p and q in polynomial time if one knows $\varphi(n)$.

A7.11.* Let n be a product of distinct primes. If d and e are positive integers such that $de - 1$ is divisible by $p - 1$ for every prime $p \mid n$, then we have $a^{de} \equiv a \pmod{n}$ for every integer a , even if a is not relatively prime with n .

A7.12. Explain why Exercise A7.11 proves that we have not to worry if in the *RSA* cryptosystem it happens that a unitary message P_i we want to send to the user B is not relatively prime with n_B .

A7.13. Prove that lines are irreducible curves. (Hint: first degree polynomials are always irreducible.)

A7.14. Consider the line R of equation (7.11). Prove that if $a \neq 0$ the projection $(u, v) \in R \rightarrow v \in \mathbb{K}$ on the y -axis is bijective, and write its inverse.

A7.15. Prove that the reducible conics are unions of two lines.

A7.16.* Prove that a curve of equation (7.12) is irreducible.

A7.17.* Prove that an irreducible conic, that is an irreducible curve defined by a degree two equation, is a rational curve, that is admits a parametric representation by rational functions.

A7.18.* Prove that the polynomial $f(x) = x^3 + ax + b$ has no multiple roots if and only if $27b^2 + 4a^3 \neq 0$. (Hint: a root of a polynomial $f(x)$ is a multiple root if and only if it is a root of the derivative of $f(x)$ too.)

A7.19. Prove that the line through the points (x_1, y_1) and (x_2, y_2) has equation (7.23) with m, n determined by (7.24). (Hint: the (non vertical) lines through (x_1, y_1) have equation $y - y_1 = m(x - x_1)$ and m can be found by imposing that the point (x_2, y_2) lies on the line.)

A7.20. Prove that if x_1, x_2, x_3 are the three roots of a degree three monic polynomial, then the degree two coefficient of the polynomial is $-x_1 - x_2 - x_3$ and the constant term is $-x_1x_2x_3$. (Hint: the polynomial may be written as $(x - x_1)(x - x_2)(x - x_3)$.)

A7.21. Let \mathbb{K} be a field and consider the curve in $\mathbb{A}_{\mathbb{K}}^2$ defined by an equation of the form $y^2 + y(mx + n) = \ell x^3 + px^2 + qx + r$ with $\ell \neq 0$. Prove that, if \mathbb{K} contains the cubic roots of each of its elements, then it is possible to reduce it in Weierstrass canonical form as in Proposition 7.9.4. (Hint: begin by changing variables by $x \rightarrow x/\sqrt[3]{\ell}$, $y \rightarrow y$, and go on as in the proof of Proposition 7.9.4.)

A7.22.* Prove that a curve defined by an equation in Weierstrass form of the kind (7.17), (7.18), (7.19) or (7.20) is singular in the algebraic closure of \mathbb{K} if and only if it does not verify the regularity hypothesis on page 373.

A7.23.* Prove that if a curve defined by an equation in Weierstrass form of the kind (7.17), (7.18), (7.19) or (7.20) is singular, then it is rational. (Hint: work in the algebraic closure of \mathbb{K} and prove that almost all the lines through a singular point intersect the curve in a unique point out of the singular point.)

A7.24. Prove that, if $27b^2 + 4a^3 = 0$, then the curve of equation (7.17) is rational.

A7.25.* Let $p = (\xi, \eta)$ be a point of the elliptic curve (7.17), with $\eta \neq 0$. Determine the intersection point different from p of the tangent line to the curve in p . (Hint: obtain y from the equation of the tangent line and substitute it in the equation of the curve; so one finds an equation of degree three in x having a double solution in $x = \xi$; the further solution of the equation is the abscissa of the required point.)

A7.26.* Explain why the line at infinity is to be considered as the tangent line to an elliptic curve in Weierstrass form in the point at infinity. (Hint: write everything in cartesian coordinates $u = x_0/x_3, v = x_1/x_3$ and consider the point O as the origin of a new plane $\mathbb{A}_{\mathbb{K}}^2$.)

A7.27.* Compute the coordinates of the point $p + q = (x_3, y_3)$ in function of the points $p = (x_1, y_1)$ and $q = (x_2, y_2)$ of an elliptic curve in each of the cases (7.17), (7.18), (7.19) and (7.20) (see page 379).

A7.28. Prove that the operation of sum of points on an elliptic curve is commutative, that is, $p + q = q + p$. (Hint: carry out the calculations, or consider the geometric definition of the group law.)

A7.29. Prove that an elliptic curve $y^2 = f(x)$ defined on the real field consists of a single arc if $f(x)$ has a single real root, while it consists of a closed arc and an open one if $f(x)$ has three real roots. In Figure 7.4 this second case is represented. Give an example of a curve for which the first case is verified. (Hint: study the sign of $f(x)$, since the curve has no point (x, y) if $f(x)$ is negative.)

In the following Exercises A7.30–A7.35 all the solutions $(x, y, z) \in \mathbb{N} \times \mathbb{N} \times \mathbb{N}$ with non-zero x, y, z of the equation $x^2 + y^2 = z^2$ are determined. These solutions are called *Pythagorean triples* as, by Pythagoras's theorem, they are the lengths of the legs and of the hypotenuse of a right triangle. A Pythagorean triple (x, y, z) is said to be *primitive* if $\text{GCD}(x, y, z) = 1$.

A7.30. Let (x, y, z) be a Pythagorean triple such that $\text{GCD}(x, y, z) = d$. Prove that $(x/d, y/d, z/d)$ is a primitive Pythagorean triple.

A7.31. Let (x, y, z) be a primitive Pythagorean triple. Prove that $\text{GCD}(x, y) = \text{GCD}(x, z) = \text{GCD}(y, z) = 1$.

A7.32. Let (x, y, z) be a primitive Pythagorean triple. Prove that one out of x and y is even, and the other is odd.

A7.33. Let r, s be positive integers such that $\text{GCD}(r, s) = 1$. Prove that if rs is a square, so are r and s .

A7.34.* Let (x, y, z) be a primitive Pythagorean triple with even y . Prove that there exist positive integers n, m , with $\text{GCD}(m, n) = 1$ and $m > n$, such that

$$x = m^2 - n^2, \quad y = 2mn, \quad z = m^2 + n^2. \quad (7.31)$$

A7.35. Verify that if (x, y, z) is a triple given by (7.31) with n, m positive integers such that $\text{GCD}(m, n) = 1$ and $m > n$, then (x, y, z) is a primitive Pythagorean triple.

A7.36. Prove that the number of solutions of the equation $x^2 = u$ in \mathbb{F}_q is $1 + \chi(u)$, where $\chi(u)$ is the quadratic character of u in \mathbb{F}_q .

A7.37. Prove that the quadratic character χ in \mathbb{F}_q satisfies $\chi(uv) = \chi(u)\chi(v)$ for all $u, v \in \mathbb{F}_q$.

A7.38. Let α', β' be the roots of the numerator of (7.30). Prove that $|\alpha| = |\beta| = \sqrt{q}$ where $\alpha = 1/\alpha', \beta = 1/\beta'$.

In the following two exercises we describe a simple polynomial probabilistic algorithm to determine points over an elliptic curve defined over \mathbb{Z}_p .

A7.39.* Let p be a prime number. Prove that there is a polynomial probabilistic algorithm that determines an integer n that *is not* a quadratic residue modulo p . (Hint: keep in mind the fact that half the elements of \mathbb{Z}_p are not quadratic residues and that computing Jacobi symbols takes a polynomial time.)

A7.40.* Let $p > 2$ be a prime number and let C be a hyperelliptic curve of equation $y^2 = f(x)$ defined over \mathbb{Z}_p . Prove that there is a polynomial probabilistic algorithm that determines a point of C . (Hint: for all $x \in \mathbb{K}$ the probability that $f(x)$ is a square is $1/2$; keep in mind the previous exercise and apply the algorithm of § 5.2.6.)

A7.41.* Explain how to use the groups defined over elliptic curves to implement the *RSA* public-key cryptosystem. (Hint: follow exactly the same steps already seen for the *RSA* system, substituting the points of an elliptic curve for the integer numbers and the operation of addition of points on the curve for the multiplication among integers.)

A7.42.* Explain how to use the groups defined over elliptic curves to implement the method to exchange private keys as described in § 7.8.1.

A7.43.* Explain how to use the groups defined over elliptic curves to implement the cryptosystem described in § 7.8.2.

B7 Computational exercises

B7.1. Which is the 2-digit numerical equivalent of **exercise**?

- (a) 0423041602081804.
- (b) 0423041702091804.
- (c) 0423041602091804.
- (d) None of the above

B7.2. Which string corresponds to the number sequence

021418040002001814,

if we have used the 2-digit numerical equivalent?

- (a) coseacoso.
- (b) coseacasa.
- (c) coseacosa.
- (d) None of the above

B7.3. Which is the binary numerical equivalent of **codes**?

- (a) 0001001100001010010010010.
- (b) 0001001110001010010010010.
- (c) 0001001100000110010010010.
- (d) 0001001110000110010010010.

B7.4. To which word does the number sequence

01100010000110100100

correspond if we have used the binary numerical equivalent?

- (a) mine.
- (b) mien.
- (c) main.
- (d) nine.

B7.5. Which is the most frequent consonant in English language texts?

- (a) N.
- (b) S.
- (c) R.
- (d) T.

B7.6. Write the vowels a, e, i, o, u in decreasing order of their frequency in a long English language text.

- (a) A, e, i, o, u.
- (b) E, i or o, o or i, a, u.
- (c) A, e or o, o or e, i, u.
- (d) None of the above.

B7.7. Determine the most frequent consonant in the following text: *analysing frequencies often is the key to a successful cryptanalysis of messages enciphered using a monoalphabetic substitution.*

- (a) N.
- (b) T.
- (c) C.
- (d) S.

B7.8. Analyse the letter frequencies in the following text: *the frequencies with which the letters appear in a short text might be very different from what we would expect*
Order the vowels a, e, i, o, u in the order of frequency in this text.

- (a) A, e, i, o, u.
- (b) E, a, i, o, u.
- (c) E, o, i, u, a.
- (d) E, i, a, o, u.

B7.9. Encipher using Caesar method, shifting each letter forward by three positions, the following message: **i am ready to attack gaul.**

- (a) L DP UHDBG WR DWWDFN JDWO.
- (b) L DP UIDGB WR DWWDFN JDWO.
- (c) L DP UHDBG WR DWWDFN JDXO.
- (d) L DP UIDGB WR DWWDFN JDXO.

B7.10. Suppose a Roman centurion received the following message sent him directly by Caesar:

DOHD NDFZAP HVZ.

After deciphering it, the centurion is very puzzled, as in the plaintext there is a patent Latin grammar error. Which is the plaintext of the message the centurion received?

- (a) **alea iacta est.**
- (b) **alea iactae est.**
- (c) **alea iactum est.**
- (d) **alea iactus est.**

B7.11. Consider the plaintext **stiff upper lip**. Which is the ciphertext in a Caesar cipher with a 13-letter shift?

- (a) **FGVSS HCCRE YVC.**
- (b) **FGUSS HCCRE YUC.**
- (c) **FGVTT HCCRE YVC.**
- (d) None of the above.

B7.12. Suppose we have received the message **XYXCXCO**, known to have been enciphered by a 10 letter shift in a Caesar cipher. Which is the plaintext?

- (a) **popcorns.**
- (b) **nonsense.**
- (c) **ascience.**
- (d) None of the above.

B7.13. Verify that the cipher with *to be or not to be that is the question* as its key phrase determines the permutation:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
19	14	1	4	17	13	7	0	8	18	16	20	2	3	5	6	9	10	11	12	15	21	22	23	24	25

B7.14. We want to encipher a message using a monoalphabetic substitution. Suppose we have chosen as our key phrase *there are more things in heaven and earth*. Which is the enciphering alphabet we get?

- (a) a b c d e f g h i j k l m n o p q r s t u v w x y z
T H E R A M O I N G S V D B C F J K L P Q U W X Y Z
- (b) a b c d e f g h i j k l m n o p q r s t u v w x y z
T H E R A M O T I N G S V D B C F J K P Q U W X Y Z
- (c) a b c d e f g h i j k l m n o p q r s t u v w x y z
T H E R A M O T I N G S D B C F J K L P Q U W X Y Z
- (d) a b c d e f g h i j k l m n o p q r s t u v w x y z
T H E A M O R I N G S D B C F J K L P Q U V W X Y Z

B7.15. Consider the monoalphabetic substitution of the previous exercise. Which ciphertext is obtained from the plaintext **than are dreamt of in your philosophy**?

- (a) **PITB TKM AKMTDP CO NB YCQK FINVCLCFIY.**

- (b) PITB TLA RLATDP CM NB YCQL FINVCLCFIY.
- (c) PITB TKA RKATDP CM NB YCQK FINVCLCFIY.
- (d) PITB TKE AKETDP CO NB YCQK FINVCLCFIY.

B7.16. Suppose we are reading a text enciphered using the monoalphabetic substitution given in the two previous exercises. If this text is LWCKR, which is the plaintext?

- (a) swear.
- (b) sword.
- (c) swore.
- (d) None of the above.

B7.17. We want to encipher a message using a monoalphabetic substitution. Suppose we have chosen as our key phrase

*Tyger Tyger, burning bright,
in the forests of the night.*

Which is the enciphering alphabet we get from this key phrase?

- (a) a b c d e f g h i j k l m n o p q r s t u v w x y z
T Y G E R B U N I H F O R S A C D J K L M Q V W X Z
- (b) a b c d e f g h i j k l m n o p q r s t u v w x y z
T Y G E R B U N F I H O S A C D J K L M P Q V W X Z
- (c) a b c d e f g h i j k l m n o p q r s t u v w x y z
T Y G E R B U N I H F O S A C D J K L M P Q V W X Z
- (d) a b c d e f g h i j k l m n o p q r s t u v w x y z
T Y G E R B U N F I H O R S A C D J K L M Q V W X Z

B7.18. Consider the monoalphabetic substitution determined by the key phrase **Tyger Tyger, burning bright, in the forests of the night**, for which we have determined the enciphering alphabet in Exercise B7.17. Which is the enciphered text we obtain from the plaintext **poem by blake**?

- (a) DCRS YX YOTFR.
- (b) DCSR XY XOTHR.
- (c) DCSR YX YOTHR.
- (d) DCRS XY XOTFR.

B7.19. Suppose we are reading a text enciphered using the monoalphabetic substitution of the previous exercise, that is, using the key phrase **Tyger Tyger, burning bright, in the forests of the night**. If this text is LIAUTOCAU, which is the plaintext?

- (a) singasong.
- (b) longsongs.
- (c) alongsong.
- (d) None of the above.

B7.20. Use the program of exercise C7.3 to encipher the message sent to Edgar Allan Poe, using the key phrase *UNITED STATES*. Check the mistakes that were made in the message as given in the text.

B7.21. Let *ALGEBRA* be the key word chosen to encipher a message using Vigenère method (see pp. 323-325). If the plaintext is **very hard to decrypt**, which is the enciphered text?

- (a) VPXC IRRD EU HFTRYAZ.
- (b) VPXC IRRD EU HFTRYAZ.
- (c) VPXC IRRD EU HFTRYAZ.
- (d) VPXC IRRD EU HFTRYAZ.

B7.22. Let again *ALGEBRA* be the key word chosen to encipher a message using Vigenère method. If the ciphertext is **I SGZF JOLGKH OZNE PDISTISPY**, which is the plaintext?

- (a) **i have solved five exercises.**
- (b) **i have solved four exercises.**
- (c) **i have solved nine exercises.**
- (d) None of the above.

B7.23. Let *HARDWORK* be the key word chosen to encipher a message using Vigenère method. If the plaintext is **all work and no play makes jack a dull boy**, which is the ciphertext?

- (a) HLC ZFKB KUD ES LZRI TABHO XRMR A WXHZ SYF
- (b) HLC ZKFB KUD ES LZRI YABHO XRMR A UXHZ SYF
- (c) HLC ZFKB KUD ER LZRI YABHO XRMR A UXHZ SYF
- (d) HLC ZKFB KUD ER LZRI TABHO XRMR A UXHZ SYF

B7.24. Let again *HARDWORK* be the key word chosen to encipher using Vigenère method. If the ciphertext is **AABHWRIVUK**, which is the plaintext?

- (a) **have a day out.**
- (b) **just one more.**
- (c) **too much work.**
- (d) None of the above.

B7.25. Encipher the message **attack today** in 4-letter blocks using the translation method in $\mathcal{P} = \mathbb{Z}_{26}^4$ described in section 7.4.1 using the key $k = 100$ (see Table 7.8 on page 340).

B7.26. Consider the plaintext **happy birthday**. Which is the text enciphered using the affine transformation with key $k = (7, 3)$?

- (a) **ADEEP KHSGAYDP.**
- (b) **ADEEM JHSGAYDM.**
- (c) **ADEEP JHSGAYDP.**
- (d) **ADEEM KHSGAYDM.**

B7.27. Suppose we receive the message **ZDB AHJ FDSCP**, knowing that it has been enciphered using the affine transformation with key $k = (7, 3)$. Which is the plaintext?

- (a) `see him later.`
- (b) `saw you early.`
- (c) `see you later.`
- (d) `saw him early.`

B7.28. Compute, if it exists, the inverse modulo 26 of the matrix

$$\begin{pmatrix} 7 & 3 \\ -5 & -10 \end{pmatrix}.$$

- (a) The inverse does not exist because the determinant of the matrix is not relatively prime with 26.
- (b) The inverse is $\begin{pmatrix} 12 & -1 \\ 7 & -11 \end{pmatrix}$.
- (c) The inverse is $\begin{pmatrix} 11 & 1 \\ 7 & -11 \end{pmatrix}$.
- (d) None of the above.

B7.29. Compute, if it exists, the inverse modulo 26 of the matrix

$$\begin{pmatrix} 19 & 13 \\ 2 & 11 \end{pmatrix}.$$

- (a) The inverse does not exist because the determinant of the matrix is not relatively prime with 26.
- (b) The inverse is $\begin{pmatrix} 11 & 13 \\ -2 & -19 \end{pmatrix}$.
- (c) The inverse is $\begin{pmatrix} -11 & 13 \\ -2 & 19 \end{pmatrix}$.
- (d) None of the above.

B7.30. We want to encipher the plaintext `computer` using an affine transformation of \mathbb{Z}_{26}^2 defined by the key $k = (A, b)$, with

$$A = \begin{pmatrix} 7 & 3 \\ -5 & -10 \end{pmatrix}, \quad b = \begin{pmatrix} 1 \\ 2 \end{pmatrix}.$$

Which is the ciphertext?

- (a) `FIAAQYCU.`
- (b) `FIABQZCU.`
- (c) `FIABQYCU.`
- (d) `FIABQZCU.`

B7.31. Consider a text enciphered using the affine transformation given in the previous exercise. If the ciphertext is `VOITJOCHGN`, which is the plaintext?

- (a) `two secrets.`
- (b) `tensecrets.`
- (c) `any secrets.`
- (d) `six secrets.`

B7.32. Trying to perform the cryptanalysis of a message we have intercepted, we are confronted with the problem of solving the following system of linear congruences:

$$\begin{cases} 19 \equiv 6a + b \pmod{26}, \\ 13 \equiv 3a + b \pmod{26}. \end{cases}$$

How many solutions does this system admit modulo 26?

- (a) One.
- (b) No one.
- (c) Infinitely many.
- (d) None of the above.

B7.33. Trying to perform the cryptanalysis of a message we have intercepted, we are confronted with the problem of solving the following system of linear congruences:

$$\begin{cases} 11 \equiv 21a + b \pmod{26}, \\ -7 \equiv -5a + b \pmod{26}. \end{cases}$$

How many solutions does this system admit modulo 26?

- (a) One.
- (b) No one.
- (c) Infinitely many.
- (d) None of the above.

B7.34. Let $S = \{1, \dots, 15\}$. Consider the function $f : S \rightarrow S$, $f(x) = 5^x \pmod{16}$. What can be said about f ?

- (a) It is bijective.
- (b) It is injective but not surjective.
- (c) It is surjective but not injective.
- (d) It is neither surjective nor injective.

B7.35. Consider the function $f : \mathbb{Z}_4 \rightarrow \mathbb{F}_5^*$, $f(x) = 4^x \pmod{5}$. What can be said about f ?

- (a) It is bijective.
- (b) It is injective but not surjective.
- (c) It is surjective but not injective.
- (d) It is neither surjective nor injective.

B7.36. Consider the function $f : \mathbb{Z}_8 \rightarrow \mathbb{F}_9^*$, $f(x) = i^x$, where i is an element of \mathbb{F}_9 such that $i^2 = -1$. What can be said about f ?

- (a) It is bijective.
- (b) It is injective but not surjective.
- (c) It is surjective but not injective.
- (d) It is neither surjective nor injective.

B7.37. Consider the function $f : \mathbb{Z}_{q-1} \rightarrow \mathbb{F}_q^*$, $f(x) = b^x$, where b is an element of \mathbb{F}_q^* . What can be said about f ?

- (a) The function f is bijective for all b .
- (b) There exists at least a value of b such that f is bijective.
- (c) There exists exactly one b such that f is bijective.
- (d) In general there is no b such that f is bijective.

B7.38. Verify that $\bar{2}$ is a generator of $U(\mathbb{Z}_{101})$. Compute the discrete logarithm $\log_2 \bar{3}$.

B7.39. Using the Baby step–giant step algorithm determine $\log_2 7$ in base 13. The answer is:

- (a) 4.
- (b) 5.
- (c) 7.
- (d) 11.

B7.40. Using the Baby step–giant step algorithm determine $\log_3 7$ in base 17. The answer is:

- (a) 4.
- (b) 11.
- (c) 7.
- (d) 16.

B7.41. Consider the sequence 1, 3, 4, 8, 13, 20. How many solutions has the knapsack problem for this sequence and $m = 34$?

- (a) None.
- (b) One.
- (c) Two.
- (d) Three.

B7.42. Consider the sequence 2, 3, 7, 8, 15, 27. How many solutions has the knapsack problem for this sequence and $m = 35$?

- (a) None.
- (b) One.
- (c) Two.
- (d) Three.

B7.43. Is the sequence 1, 5, 8, 15, 30, 60 superincreasing?

- (a) Yes.
- (b) No, because $2 \cdot 5 > 8$ and $2 \cdot 8 > 15$.
- (c) No, because $60 \leq 1 + 5 + 8 + 15 + 30$.
- (d) None of the above.

B7.44. Is the sequence 1, 2, 5, 9, 17, 45 superincreasing?

- (a) Yes.
- (b) No, because $2 \cdot 5 > 9$.
- (c) No, because $17 \leq 1 + 2 + 5 + 9$.
- (d) None of the above.

B7.45. Consider the superincreasing sequence 1, 2, 5, 11, 22, 44, 88. How many solutions has the corresponding knapsack problem for $m = 147$?

- (a) None.
- (b) The only solution is $x_1 = x_2 = x_4 = x_6 = x_7 = 1$ and $x_3 = x_5 = 0$.
- (c) The only solution is $x_3 = x_4 = x_6 = x_7 = 1$ and $x_1 = x_2 = x_5 = 0$.
- (d) None of the above.

B7.46. Consider the superincreasing sequence 1, 4, 7, 13, 28, 54. How many solutions has the corresponding knapsack problem for $m = 76$?

- (a) None.
- (b) The only solution is $x_1 = x_3 = x_4 = x_6 = 1$ and $x_2 = x_5 = 0$.
- (c) The only solution is $x_1 = x_2 = x_3 = x_6 = 1$ and $x_4 = x_5 = 0$.
- (d) None of the above.

B7.47. Consider the knapsack problem cipher. We have chosen the superincreasing sequence 1, 3, 6, 12, $m = 29$ and $w = 10$. Which is the public key we have to publish to have people send us enciphered messages?

- (a) It is the sequence 10, 1, 2, 4.
- (b) It is the sequence 10, 2, 4, 8.
- (c) It is the sequence 10, 2, 3, 6.
- (d) None of the above.

B7.48. Let 1, 2, 5, 20 be our superincreasing sequence, $m = 43$ and $w = 25$. Which is the public key we have to publish to have people send us enciphered messages?

- (a) It is the sequence 25, 7, 39, 29.
- (b) It is the sequence 25, 7, 37, 27.
- (c) It is the sequence 25, 7, 39, 27.
- (d) None of the above.

B7.49. Consider the example of knapsack problem cipher illustrated in Table 7.10 on page 350. If the plaintext to be sent is **otto**, which is the numerical equivalent of the ciphertext?

- (a) 73 17 58 41 58.
- (b) 73 17 50 41 58.
- (c) 73 34 50 41 58.
- (d) 73 34 58 41 58.

B7.50. Consider again the example in Table 7.10 on page 350. If the plaintext to be sent is **casa**, which is the numerical equivalent of the ciphertext?

- (a) 32 0 7 34 0.
- (b) 32 0 17 34 0.
- (c) 32 0 7 0 17.
- (d) 32 0 7 17 0.

B7.51. To access the *RSA* system, Blanche wants to publish the enciphering key (7927, 37), but the system does not accept this key. Why?

- (a) Because 7927 is too large.
- (b) Because 37 is too small.
- (c) Because 7927 is not the product of two prime numbers.
- (d) Because 37 is not relatively prime with $\varphi(7927)$.

B7.52. To access the *RSA* system, Ariadne wants to publish the enciphering key (9991, 119), but the system does not accept this key. Why?

- (a) Because 9991 is too large.
- (b) Because 119 is too small.
- (c) Because 9991 is not the product of two prime numbers.
- (d) Because 119 is not relatively prime with $\varphi(9991)$.

B7.53. Consider the user directory (7.8) on page 352. A user of the *RSA* system wants to send the message **baba** to the user *B*, Beatrix. Which enciphered message does Beatrix receive?

- (a) $C_1 = 9, C_2 = 9$.
- (b) $C_1 = 999, C_2 = 9$.
- (c) $C_1 = 999, C_2 = 999$.
- (d) None of the above.

B7.54. Another user of the *RSA* system wants to send Beatrice a message. If the plaintext is **coda**, which enciphered message does Beatrix receive?

- (a) $C_1 = 31, C_2 = 243$.
- (b) $C_1 = 546, C_2 = 243$.
- (c) $C_1 = 546, C_2 = 576$.
- (d) $C_1 = 31, C_2 = 576$.

B7.55. Consider again the user directory (7.8) on page 352. Beatrix has received the following message:

$$C_1 = 31, \quad C_2 = 722, \quad C_3 = 272.$$

Which is the numerical equivalent of the plaintext?

- (a) 0214 0308 0204.
- (b) 0214 0300 0208.
- (c) 0214 0308 0208.
- (d) None of the above.

B7.56. Beatrix has received the following message:

$$C_1 = 243, \quad C_2 = 722.$$

Which is the numerical equivalent of the plaintext?

- (a) 0308 0200.
- (b) 0300 0308.
- (c) 0200 0308.
- (d) None of the above.

B7.57. Beatrix has received the following message:

$$C_1 = 546, \quad C_2 = 722, \quad C_3 = 999.$$

Which is the numerical equivalent of the plaintext?

- (a) 0314 0308 0208.
- (b) 0308 0208 0314.
- (c) 0308 0314 0208.
- (d) 0314 0208 0308.

B7.58. Verify that the real curve of equation $x^3 + xy^2 - x - 3x^2 - 3y^2 + 3 = 0$ is not singular, but it is so as a curve over \mathbb{C} .

B7.59. Give an example of a curve that is not singular over \mathbb{Q} but is singular over \mathbb{R} .

B7.60. Consider the elliptic curve over \mathbb{R} of equation $y^2 = x^3 - x$. Let $p = (-1, 0)$ and $q = (2, -\sqrt{6})$. Which are the coordinates of $p + q$?

- (a) $p + q$ is the point at infinity O .
- (b) $p + q = (-1/3, 2\sqrt{6}/9)$.
- (c) $p + q = (-1/3, -2\sqrt{6}/9)$.
- (d) None of the above.

B7.61. Consider the elliptic curve over \mathbb{R} of equation $y^2 = x^3 - x$. Let $p = (1, 0)$ and $q = (2, \sqrt{6})$. Which are the coordinates of $p + q$?

- (a) $p + q$ is the point at infinity O .
- (b) $p + q = (3, 2\sqrt{3})$.
- (c) $p + q = (3, -2\sqrt{3})$.
- (d) None of the above.

B7.62. Consider the elliptic curve over \mathbb{R} of equation $y^2 = x^3 - x$. Let $p = (-1, 0)$. Which are the coordinates of $2p = p + p$ (in the group law on the curve)?

- (a) $2p$ is the point at infinity O .
- (b) $2p = (-1, 0)$.
- (c) $2p = (0, 0)$.
- (d) None of the above.

B7.63. Consider the elliptic curve over \mathbb{R} of equation $y^2 = x^3 - x$. Let $p = (2, -\sqrt{6})$ and $q = (0, 0)$. Which are the coordinates of $p + q$?

- (a) $p + q$ is the point at infinity O .
- (b) $p + q = (-1/2, \sqrt{6}/4)$.
- (c) $p + q = (-1/2, -\sqrt{6}/4)$.
- (d) None of the above.

B7.64. Let C be the elliptic curve of equation $y^2 = x^3 - x$ over the field \mathbb{F}_7 . Are $p = (1, 0)$ and $q = (-2, -1)$ points of C ?

- (a) Both p and q are points of C .
- (b) The point p is on C , but q is not.

- (c) The point q is on C , but p is not.
- (d) Neither p nor q belong to C .

B7.65. Let C be the elliptic curve of equation $y^2 = x^3 - x$ over the field \mathbb{F}_5 . Are $p = (2, 1)$ and $q = (-2, 2)$ points of C ?

- (a) Both p and q are points of C .
- (b) The point p is on C , but q is not.
- (c) The point q is on C , but p is not.
- (d) Neither p nor q belong to C .

B7.66. Prove that the elliptic curve of equation $y^2 + y = x^3 + 1$ has three points over \mathbb{Z}_2 , including the point at infinity.

B7.67. How many points has the curve $y^2 = x^3 - x$ over \mathbb{F}_7 ?

- (a) 3.
- (b) 6.
- (c) 8.
- (d) 10.

B7.68. How many points has the curve $y^2 + y = x^3$ over \mathbb{F}_8 ?

- (a) 4.
- (b) 5.
- (c) 7.
- (d) 9.

C7 Programming exercises

C7.1. Write a program that implements any Caesar cipher, that is, given in input a number n , with $1 \leq n \leq 25$, and a text, it outputs the text enciphered by shifting each letter by n positions. Then write a program that deciphers a message so enciphered.

C7.2. Write a program that, given a text as its input, outputs a frequency table of the letters appearing in the text.

C7.3. Write a program that, given as input a key word and a text, outputs the text enciphered using Vigenère method (see pp. 323-325) with the given key word.

C7.4. Write a program that, given in input a text enciphered with Vigenère method and the key word, outputs the plaintext.

C7.5. Write a program that computes the inverse of a square matrix modulo a positive integer, if it exists.

C7.6. Write a program that, given in input a text and two integers a, b , outputs the text enciphered with the affine transformation $C_{a,b} : \mathbb{Z}_{26} \rightarrow \mathbb{Z}_{26}$, $C_{a,b}(P) = aP + b \pmod{26}$.

C7.7. Write a program that, given in input a text, an $s \times s$ square matrix A and a vector b of length s , outputs the text enciphered with the affine transformation $C_{A,b} : \mathbb{Z}_{26}^s \rightarrow \mathbb{Z}_{26}^s$, $C_{A,b}(p) = Ap + b \pmod{26}$, where p is a vector of length s .

C7.8. Write a program that computes discrete logarithms using the Baby step–giant step algorithm.

C7.9. Write a program that verifies whether an n -integer sequence is superincreasing or not.

C7.10. Write a program that generates a superincreasing n -integer sequence.

C7.11. Write a program that, given in input a superincreasing sequence a_1, \dots, a_n and an integer m , outputs the solution to the corresponding knapsack problem, if it exists. (Hint: use the algorithm described in the text.)

C7.12. Write a program that, given in input an integer N , outputs: (1) a superincreasing sequence a_1, \dots, a_N , an integer $m > 2a_N$ and an integer w relatively prime with n (the *private* data of a user X); (2) the sequence $b_j = wa_j \pmod{m}$ (the *public* key of user X).

C7.13. Write a program that, given in input a sequence (b_j) , constituting the public key in a Merkle–Hellman system, and a plaintext, outputs the text enciphered with the b_j s to be sent to the user X .

C7.14. Write a program that, given in input a ciphertext and the deciphering private key, outputs the plaintext. (Hint: use the program solving the knapsack problem for a superincreasing sequence.)

C7.15. Write a program that randomly generates a prime number with a given number of digits. (Hint: use the algorithm described in Remark 7.7.1.)

C7.16. Write a program that, given in input a positive integer N , outputs a pair of integers (n, e) such that n is the product of two prime numbers p, q each having N digits, and e is relatively prime with both $p - 1$ and $q - 1$ (so we may use the pair (n, e) as public key to use an *RSA* system).

C7.17. Write a program that, given in input a plaintext and the public key (n, e) of a user A , outputs the ciphertext to be sent to A using the *RSA* system.

C7.18. Write a program that, given in input the ciphertext and the private information $n = pq$ in an *RSA* system, outputs the deciphered text.

C7.19. Write a program that finds points on an elliptic curve on a finite field of characteristic different from 2.

C7.20. Write a program that, given in input a prime number p (sufficiently small) and the equation of an elliptic curve over \mathbb{Z}_p , computes how many points of the plane lie on the curve. (Hint: proceed by trial and error for all the values in \mathbb{Z}_p of x in the equation in Weierstrass form.)

C7.21. Write a program that, given in input the coordinates of two points p and p' of an elliptic curve over a finite field \mathbb{F}_q , outputs the coordinates of the point $p + p'$. (Hint: use the equations given in the text.)

C7.22. Write a program that, given in input the coordinates of a point p of an elliptic curve defined over a finite field \mathbb{F}_q , determines the order of p .

C7.23. Write a program that factors a number using Pollard's $p - 1$ algorithm.

C7.24. Write a program that computes $[\sqrt{n}]$ for an integer n .

Elementary Number Theory, Cryptography and Codes

Baldoni, M.W.; Ciliberto, C.; Piacentini Cattaneo, G.M.

2009, XVI, 522 p. 10 illus., Softcover

ISBN: 978-3-540-69199-0