Chapter 3

Number theory and cryptography

3.1 Cryptography: Overview

3.1.1 Aim of Cryptography

The need to protect information is as old as civilisation itself. At first used only by the military and diplomats, it has gained much commercial use with the appearance of computer networks. For instance, consider a financial transaction which is accomplished electronically. Clearly, the user of such a system will demand that any data exchanged during such a transaction be confidential and the provider of the system will want to ensure that fraudulent use of the system is excluded. A further concern that has to be addressed is the issue of authentification (how can I be certain that a message was really written by the person who claims to be the author). This requires the construction of the equivalent of a signature.

The principle of cryptography is represented in Fig. 3.1. During the encryption (or enciphering) operation, the *plaintext* P is transformed by a function E with *key* K into the *ciphertext* $C = E_K(P)$. This ciphertext is then sent to the receiver who applies the decryption algorithm D, which is the inverse of the encryption: $P = D(C) = D(E_K(P))$. We suppose that an "intruder" is listening and can reproduce the whole ciphertext. He does not, however, know the encryption key although it is assumed that the specific functions used are known. This assumption is known as *Kerckhoffs' thesis*. Sometimes the intruder not only listens in on the communication channel (passive intrusion), but can alter the messages or insert his own messages into the communication channel (active intrusion). The art of composing ciphertexts is known as *cryptography*, the art of breaking them is *cryptanalysis*.

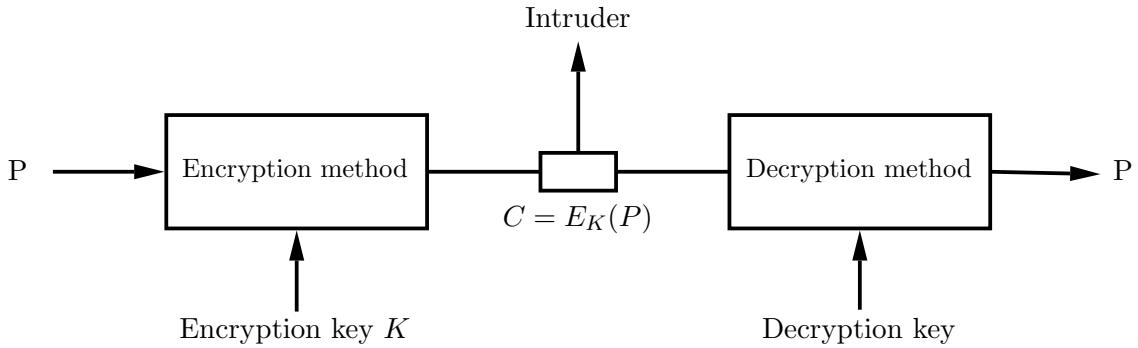

Figure 3.1: Model of a cryptosystem.

Depending on the cryptanalyst's knowledge, we can distinguish between various types of attacks. If he only has access to samples of ciphertext, but does not know the corresponding plaintext, then we speak of a *ciphertext-only attack*. Clearly a cipher must be designed to be secure against such an attack, since by assumption of our model the intruder always has access to the ciphertext. If the cryptanalyst can analyse pairs of plaintext and corresponding ciphertext, this is known as a *known plaintext attack*. Such an attack is possible if the intruder at some point gains access to past records of plaintext and the corresponding transmitted message. Clearly,

as we cannot be certain that this will never happen, our system must also be able to resist this kind of attack. Finally, sometimes the cryptanalyst is lucky enough to be able to obtain the ciphertext for any plaintext of his choice (imagine for example that he gains temporary access to the encryption machine, but cannot see the key); this is called a *chosen plaintext attack*.

The most important reason for Kerckhoffs' assumption, *i.e.,* that we assume that the encryption algorithm is known, is the following. History has shown that a secret does not stay a secret for long once it is shared by several people, so the design is unlikely to remain unknown. Although it would be long and costly to change the encryption algorithm each time the method is compromised, it is not a problem to frequently change the key. The basic encryption model is thus composed of a constant and known general encryption method, parametrized by a secret and easily modifiable key. Given a cryptosystem, we would like to have the guarantee that it cannot (easily) be broken. The strongest form of such a guarantee are the so-called *unconditionally secure ciphers*. These are ciphers which cannot be broken even if we assume that the attacker has infinite computational power. Slightly weaker are the so-called *computationally secure ciphers*, which are cipher which could in principle be broken by a cryptanalyst, but breaking them would require computational resources many orders of magnitude beyond what is available.

3.2 Symmetric Ciphers

We shall now present a first category of ciphers for which the same secret key is used for encryption and decryption. Hence they are named symmetric ciphers. These ciphers require communicating the key to the receiver via a secure channel as shown in Fig. 3.2. We shall later study asymmetric ciphers, which don't use the same key for encryption and decryption.

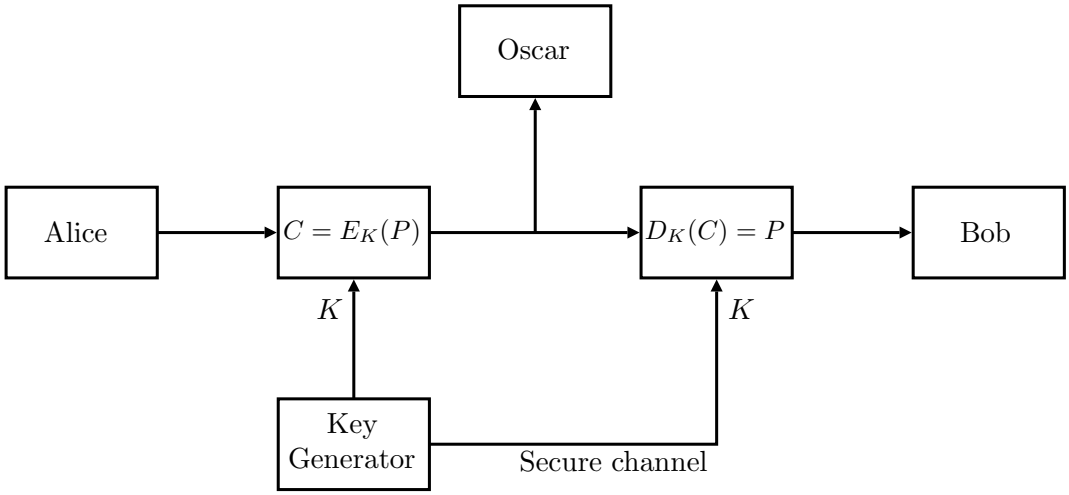

Figure 3.2: Symmetric Cipher.

3.2.1 Substitution Ciphers

Julius Caesar used an encryption mechanism which consisted in a rotation of all letters of the alphabet by three positions. So a became d, b became e, ... and z became c. Caesar's encryption method can be generalised to allow a rotation of k letters, instead of always using 3 letters. In this case, k is the key of the general encryption method which consists in a rotation of the alphabet.

Example 1. *In the movie* Space Odyssey 2001*, the name of the computer is HAL, which is an actual ciphertext. What is the plaintext? And what is the key k?*

Clearly, the rotation cipher is not very secure. Since by assumption (Kerckhoffs' thesis) the cryptanalyst is aware of the fact that we use a rotation cipher, there are only 26 possible keys and it is trivial to try them all to find the correct key.

A more secure generalisation is to replace each letter of the alphabet in the plaintext by another letter, without respecting a rotation mapping between the two alphabets. This general system is called *monoalphabetic substitution*, the key being the correspondence table between the plaintext and ciphertext alphabets. For example, we could take this key:

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

For an alphabet of D characters, there are $D!$ possible keys. Since $26! \sim 10^{26}$, a very large number, it seems that this cipher is quite secure. Unfortunately, this is not quite so. The cryptanalyst greatly eases his task by looking at the distribution of letter frequencies in the ciphertext. In English language, for example, the following table shows the corresponding letter frequencies.

letter	frequency[%]	letter	frequency[%]
E	13	M	2
T	9	W	2
A	8	F	2
O	8	G	2
I	7	Y	2
N	7	P	2
S	6	B	2
H	6	V	1
R	6	K	1
D	4	J	0
L	4	X	0
C	3	Q	0
U	3	Z	0

We see that the most frequent letters are e and t which the cryptanalyst can then try to match to the two most frequent letters in the ciphertext. He will then probably find many triplets of the form tXe, which strongly suggest that X corresponds to h. By continuing in this manner letter by letter, he can find the encryption key reasonably quickly.

Furthermore, he can sometimes guess certain words depending on the context. For instance, if the message concerns a stock-market transaction, the message may well contain the words "share" or "trading"...

In order to make the cryptanalist's task more difficult, it is thus necessary to "hide" the distribution of letter frequencies in such a way that letters like e, a and t be not so easy to spot. One way of doing this is to introduce several cyclically shifted alphabets, which gives a Vigenère cipher. This is an example of encryption using *polyalphabetic substitution*:

original alphabet	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
row 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
row B	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
row C	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
row D	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
row E	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
row F	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
row G	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
row H	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
row I	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
row J	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
row K	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
row L	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
row M	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
row N	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
row O	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
row P	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
row Q	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
row R	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
row S	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
row T	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
row U	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
row V	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
row W	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
row X	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
row Y	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
row Z	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

As in a monoalphabetic cipher, this polyalphabetic cipher also has a key, which is usually a short, easy to remember word, like $GOODBYE$ (instead of the complete table of 26 letters

in the monoalphabetic case). The key is constantly repeated underneath the plaintext, and indicates which row of the previous table should be used for the encryption. In the following example, v is encoded with the alphabet of row G, so it becomes B, and so on:

plaintext:	v	i	g	e	n	e	r	e	s	c	i	p	h	e	r
key:	G	O	O	D	B	Y	E	G	O	O	D	B	Y	E	G
ciphertext:	B	W	U	H	O	C	V	K	G	Q	L	Q	F	I	X

Although it is much more secure than a monoalphabetic substitution cipher, a polyalphabetic substitution cipher can still be broken by a ciphertext-only attack. The trick lies in guessing the length of the key.

A famous example of a cryptosystem based on polyalphabetic substitution is the ENIGMA machine used by the German forces and their allies during the second world war. The substitution was made using three rotors which each had 26 positions. The initial positions of the rotors where encoded in the head of the message, which was a weakness exploited by British, French and Polish cryptanalysts. (See *http://www.bletchleypark.org.uk/* for more details.)

3.2.2 Transposition Ciphers

A *transposition cipher* changes the order of the letters of the messages, but does not "disguise" them.

Example 2. *To illustrate such a cipher, we will encrypt some plaintext with the key MEGABUCK:*

plaintext: *pleasetransferonemilliondollarsto*
myswissbankaccountsixtwotwo

key:	*M*	*E*	*G*	*A*	*B*	*U*	*C*	*K*
	p	*l*	*e*	*a*	*s*	*e*	*t*	*r*
	a	*n*	*s*	*f*	*e*	*r*	*o*	*n*
	e	*m*	*i*	*l*	*l*	*i*	*o*	*n*
	d	*o*	*l*	*l*	*a*	*r*	*s*	*t*
	o	*m*	*y*	*s*	*w*	*i*	*s*	*s*
	b	*a*	*n*	*k*	*a*	*c*	*c*	*o*
	u	*n*	*t*	*s*	*i*	*x*	*t*	*w*
	o	*t*	*w*	*o*	*a*	*b*	*c*	*d*

ciphertext: *AFLLSKSOSELAWAIATOOSSCTCLNMOMANT*
ESILYNTWRNNTSOWDPAEDOBUOERIRICXB

This cipher is parametrized by a key which is a word or a sentence in which each letter only appears once. The plaintext is written as a succession of rows of the same length as the key

(the message is completed with random letters if necessary). The ciphertext is read column by column in alphabetic order of the letters making up the key.

Even if the cryptanalyst does not know the used cipher, in the case of the transposition cipher, it is not very hard to guess: this is because the letter frequency distribution in the ciphertext remains the same as in the plaintext. He must then guess the length of the key that was used and afterward find the correct arrangement of the columns.

Another example of a transposition cipher is that of *permutation* which cuts up the plaintext into blocks of length d where d is the length of the key. The d characters are then permuted following the alphabetical order of the letters making up the key.

Example 3. *In the case of the previous example ($d = 8$), we have:*

key:	*MEGABUCK*			
plaintext:	*pleasetr*	*ansferon*	*emillion*	*dollarst*
	omyswiss	*bankacco*	*untsixtw*	*otwo*
ciphertext:	*ASTLERPE*	*FEONSNAR*	*LLOMINEI*	*LASOLTDR*
	SWSMYSOI	*KACANOBC*	*SITNTWUX*	*OTWO*

Even if the length d of the key is known to the cryptanalyst, he must try the $d!$ possible permutations of the d characters making up the key.

3.2.3 Entropy of a language

In order to be able to asses the security of a cipher, we need to talk a little bit about the "entropy" of a language. Think of a page filled with characters and the English language. How many different pages can be composed which are grammatically correct? Let this set of grammatically correct pages be $\mathcal{E}$ and its cardinality be $|\mathcal{E}|$. Let the total set of all possible pages (grammatically correct or not) be denoted by $\mathcal{P}$ with cardinality $|\mathcal{P}|$.

Now regard two pages. How many possible different pages do we have in total? Clearly there are $|\mathcal{P}|^2$. How many are grammatically correct? We expect roughly $|\mathcal{E}|^2$. Actually, we expect slightly more because of boundary effects (we are now allowed to have sentences which cross the two pages).

Now look at the general case of n pages. There are $|\mathcal{P}|^n$ possible pages and roughly $|\mathcal{E}|^n$ of them are grammatically correct.

It is easier to work with logarithms: define $N(n)$ as the number of n page documents which are grammatically correct and let $M(n) = \log_2 N(n)$.

Definition 1. *Let H_L be the real number*

$$H_L \doteq \lim_{n\to\infty} \frac{1}{n} M(n).$$

We call H_L the entropy of the language (here measured per page). This entropy is just another way of saying how many pages are grammatically correct.

This means that the number of correct n-page documents is roughly 2^{nH_L}, out of a possible $2^{n\log_2(|\mathcal{P}|)}$ such documents.

Suppose now that instead of pages we talk about characters. In the roman alphabet, there are 26 possible characters. The number of possible strings of n characters is therefore $2^{n\log_2(26)}$. You will remember from Chapter 2 that entropy per character can also be seen as the amount of information conveyed per character. If we take a language which allows all strings, then there are $2^{n\log_2(26)}$ possible strings of length n. This gives the language an entropy per character of $\log_2(26) = 4.7$, which is the maximal entropy on a 26 character language.

If we take the language which allows only strings of consecutive a's, there is only one possible string of length n giving an entropy of 0. No information is carried, because we know that each character is an a. The entropy on a character set $\mathcal{C}$ of $|\mathcal{C}|$ characters is thus bounded between 0 and $\log_2(|\mathcal{C}|)$, with a higher entropy showing that there is a higher number of legal (grammatically correct) strings.

Example 4. *English has an entropy per character of about $H_{English} = 1.5$. This means that there are about $2^{1.5n}$ grammatically correct strings of n characters. Put in another way: consider strings of 100 characters. There are $26^{100} \simeq 10^{141}$ such strings. Out of those, $2^{150} \simeq 10^{45}$ are grammatically correct.*

Assume now that we are given a ciphertext C and that the keys for this cryptosystem are taken from the set $\mathcal{K}$. Assume that the number of keys is $2^{H(\mathcal{K})}$ where $H(\mathcal{K}) = \log_2(|\mathcal{K}|)$. A brute force attack on the cipher consists in trying to decrypt C with each possible key and to see which decryptions result in a "meaningful" plaintext. We decrypt C with each of these keys. More precisely, the messages are written in a character set $\mathcal{C}$. How many of them are meaningful if we know that the plaintext is written in language L?

Since there are $2^{H(\mathcal{K})}$ possible keys, there are $2^{H(\mathcal{K})}$ potential plaintexts. We assume that for each key which is not the correct key, we end up with a plaintext which is evenly distributed over the $|\mathcal{C}|^n$ possible strings. Therefore, the probability that such a potential plaintext turns out to be grammatically correct is equal to

$$\frac{\#\text{ of correct plaintexts}}{\#\text{ of possible plaintexts}} = \frac{2^{nH_L}}{2^{n\log_2(|\mathcal{C}|)}}.$$

We conclude that the expected number of meaningful decryptions is

$$\frac{2^{H(\mathcal{K})}\ 2^{nH_L}}{2^{n\log_2(|\mathcal{C}|)}} = 2^{n(\frac{1}{n}H(\mathcal{K})+H_L-\log_2(|\mathcal{C}|))}$$

If this number ~ 1 then we expect to find only a single meaningful decryption. This means that, in principle, we are able to break the cipher. On the other hand, if there were many which were meaningful, it might no longer be possible to determine which of these was the correct decryption.

For the number of meaningful decryptions to be 1, we need to have

$$\frac{1}{n}H(\mathcal{K}) + H_L - \log_2(|\mathcal{C}|) = 0.$$

We can now define the concept of *unicity distance*. This is the length of the plaintext for which the expected number of meaningful decryptions is only one.

$$n = \frac{H(\mathcal{K})}{\log_2(|\mathcal{C}|) - H_L}.$$

Example 5. *In the case of a substitution cipher and a plaintext written in english, we have*

$$\begin{aligned} \log_2 |\mathcal{K}| &= \log_2(26!) = 88 \\ H_{English} &= 1.5 \\ \log_2(|\mathcal{C}|) &= \log_2(26) = 4.7 \\ n &= \frac{H(\mathcal{K})}{\log_2(|\mathcal{C}|) - H_L} = \frac{88}{4.7-1.5} = 28. \end{aligned}$$

This means that we expect strings of 28 characters or more to have a unique meaningful key.

3.3 Elements of Number Theory

We shall move on to public key cryptosystems in the next section, but before that, we need to look at some topics in number theory which our public key cryptosystems will rely on.

3.3.1 Groups, Semi-Groups, and the Integers

We designate by $\mathbb{Z}$ the set of integers (both positive and negative). All arithmetic lessons in primary school start with the fundamental operations which can be applied to integers: addition and multiplication. Let us recall the basic properties related to these operations. If a, b and c are elements of $\mathbb{Z}$, addition has the following properties:

- Closure: $a + b \in \mathbb{Z}$
- Associativity: $a + (b + c) = (a + b) + c$
- Identity: $a + 0 = a$
- Inverse: $a + (-a) = 0$
- Commutativity: $a + b = b + a$.

A set (in our case $\mathbb{Z}$), together with a binary operation (here $+$) which fulfills the above properties (closure, associativity, identity element, inverse, commutativity) is called an *abelian group* or *commutative group*. Multiplication has the following properties:

- Closure: $a \cdot b \in \mathbb{Z}$
- Associativity: $a \cdot (b \cdot c) = (a \cdot b) \cdot c$
- 1 is the Identity: $a \cdot 1 = a$
- Commutativity: $a \cdot b = b \cdot a$

Unfortunately, only 1 and -1 have a multiplicative inverse in $\mathbb{Z}$. Therefore, $\mathbb{Z}$ together with multiplication is not a commutative group but only a *commutative semi-group*.

Further, multiplication is distributive with respect to addition, that is

$$\begin{aligned} a \cdot (b + c) &= a \cdot b + a \cdot c \\ (a + b) \cdot c &= a \cdot c + b \cdot c. \end{aligned}$$

A set (in our case $\mathbb{Z}$) together with two binary operations (here $+$ and $\cdot$) which form an abelian group and an abelian semi-group, respectively, and fulfill the distributive law is called a *commutative ring*.

The substraction operation $a - b$ is simply the addition operation $a + (-b)$ where b is replaced by its additive inverse. The division operation, however, is not always defined because the

multiplicative inverse of a is not an integer (unless $a = \pm 1$). Nevertheless, if $a = bc$ for two integers b and c, we say that b (or c) divides a, or that b (or c) is a divisor of a, and we write $b|a$ (or $c|a$). A *prime number* is a positive integer other than 1 which has no other divisors than 1 and itself.

3.3.2 Euclidean Division

Let a and b be integers. If b is not zero, then there exist two integers, the *quotient* q and the *remainder* r, such that

$$a = bq + r \qquad \text{and} \qquad 0 \leq r < |b|.$$

The calculation of q and r is called the *Euclidian division* of a by b.

3.3.3 Greatest Common Divisor (GCD)

Let a and b be two integers. The set of positive integers which divide both a and b is finite. Further, it is non-empty since 1 divides any integer. Consequently, this set has a member which is greater than or equal to all others: this is the so called *greatest common divisor* of a and b, written as $\gcd(a, b)$. If $\gcd(a, b) = 1$, a and b are *relatively prime* (or *coprime*).

Some basic properties of the GCD are:

1. $\gcd(a, b) = \gcd(b, a)$, by symmetry of the definition,
2. $\gcd(a, 0) = a$, as a is the largest divisor of a and any integer divides 0,
3. $\gcd(a, b) = \gcd(a, b + ca)$ for any $c \in \mathbb{Z}$.

This gives rise to an efficient way of computing $\gcd(a, b)$ by a judicious choice of c in property 3. Let's assume, without loss of generality (because of property 1), that $|b| \geq |a|$. From Section 3.3.2, we know that we can write $b = aq + r$, where $0 \leq r < |b|$ which yields $r = b - qa$. This allows us to write $\gcd(a, b) = \gcd(a, b - qa) = \gcd(a, r)$. By successively replacing the greater term by the remainder of the Euclidean division, we will eventually reduce our expression to $\gcd(a, b) = \gcd(r_n, r_{n-1}) = \gcd(r_n, 0) = r_n$:

$$\begin{array}{rcll} b & = & aq_1 + r_1 & \gcd(a,b) = \gcd(a, b - aq_1) \quad |a| < |b| \\ & & & = \gcd(a, r_1) \quad r_1 < |a| \\ a & = & r_1 q_2 + r_2 & = \gcd(a - r_1 q_2, r_1) \\ & & & = \gcd(r_2, r_1) \quad r_2 < r_1 \\ r_1 & = & r_2 q_3 + r_3 & = \gcd(r_2, r_3) \quad r_3 < r_2 \\ & \vdots & & \vdots \quad \vdots \\ r_{n-2} & = & r_{n-1} q_n + r_n & = \gcd(r_n, r_{n-1}) \quad r_n < r_{n-1} \\ r_{n-1} & = & r_n q_{n+1} + 0 & = \gcd(r_n, 0) \quad 0 = r_{n+1} < r_n \\ & & & = r_n. \end{array}$$

We know that this algorithm will terminate because r_n is a decreasing positive series: $|b| > |a| > r_1 > r_2 > \ldots > r_n \geq 0$.

3.3.4 Extended Euclidean Algorithm

Theorem 1. *For any integers a and b, there exist integers α and β such that*

$$\alpha a + \beta b = \gcd(a, b). \tag{3.1}$$

.

This is known as *Bézout's identity* and is useful, in particular, for solving *Diophantine equations* (equations with integer coefficients whose solutions are integers).

In order to find the values α and β, we can extend the previous algorithm somewhat, hence the name *extended Euclidean algorithm*. Working backwards, we can start with $\gcd(a, b) = r_n$ and substitute each r_k with $r_{k-2} - r_{k-1}q_k$ for k between 2 and n, eventually replacing r_1 by $b - aq_1$ so as to be left with a *linear combination* of a and b.

$\gcd(a, b)$	$=$	r_n	
	$=$	$r_{n-2} - r_{n-1}q_n$	eliminated r_n
	$=$	$r_{n-2} - (r_{n-3} - r_{n-2}q_{n-1})q_n$	eliminated r_{n-1}
	$=$	$(-q_n)r_{n-3} + (1 + q_{n-1}q_n)r_{n-2}$	grouped like terms
	$=$	$r_{n-3}\rho_a + r_{n-2}\rho_b$	replaced complex terms by coefficients
	$=$	$r_{n-4}\pi_a + r_{n-3}\pi_b$	eliminated r_{n-2}
	$\vdots$		
	$=$	$\delta_a r_1 + \delta_b r_2$	
	$=$	$\gamma_a a + \gamma_b r_1$	eliminated r_2
	$=$	$\alpha a + \beta b$	eliminated r_1.

As we can see, each remainder r_k can be expressed as a linear combination of a and b. This is true in particular for $r_n = \gcd(a, b)$.

Example 6. *Let's use this method to calculate the missing values in the following Bézout's identity:* $\alpha 120 + \beta 23 = \gcd(120, 23)$.

a	b	q_n	r_n
120	23	5	$5 = \quad 1 \cdot 120 - 5 \quad \cdot 23$
5	23	4	$3 = -4 \cdot 120 + 21 \cdot 23$
5	3	1	$2 = \quad 5 \cdot 120 - 26 \cdot 23$
2	3	1	$1 = -9 \cdot 120 + 47 \cdot 23$

Which shows that $-9 \cdot 120 + 47 \cdot 23 = \gcd(120, 23) = 1$.

3.3.5 Congruences

Let a and b be two integers. We say that a *is congruent to* b *modulo* m and write

$$a \equiv b \pmod{m} \tag{3.2}$$

if and only if m divides $a - b$. The expression (3.2) is called *congruence* and m is its *modulus*. The congruency (3.2) thus implies that

$$a = b + xm \tag{3.3}$$

for a certain integer x.

Note that, if r is the remainder of the Euclidian division of a by m, we have

$$a \equiv r \pmod{m}$$

and r is known as the *residue* of a modulo m. We can easily show that $a \equiv b \pmod{m}$ if and only if their residues modulo m are the same.

Similarly, the modulus m being fixed, we can show that the relation between a and b as defined by (3.2) is a relation of equivalence on $\mathbb{Z}$ because it is

- reflexive: $a \equiv a \pmod{m}$,
- symmetrical: if $a \equiv b \pmod{m}$ then $b \equiv a \pmod{m}$,
- and transitive: if $a \equiv b \pmod{m}$ and $b \equiv c \pmod{m}$ then $a \equiv c \pmod{m}$

Consequently, this relation defines *equivalence classes*, obtained from any integer a by adding all the multiples of m. We denote these equivalence classes by $|a|_m$ and call a a *representative* of $|a|_m$. We can easily show that each equivalence class contains a unique integer r between 0 and m: $0 \leq r < m$, which is the residue of all the elements of the class $|r|_m$. There are therefore m equivalence classes which are $|0|_m$, $|1|_m$,..., $|m-1|_m$. The set of these m equivalence classes, $\mathbb{Z}_m$, is the set of *integers modulo m*.

3.3.6 Operations Modulo m

Let us consider two classes modulo m; a is a representative of the first and b of the second. Let us define the *sum* of these two classes as the class of $a + b$ and the *product* as the class of ab:

$$|a|_m + |b|_m \doteq |a + b|_m \qquad \textit{and} \qquad |a|_m \cdot |b|_m \doteq |ab|_m$$

Let us first verify that the above definitions do not depend on the particular representative chosen for each equivalence class. If a and a' are two representatives of $|a|_m$ and b and b' are two representatives of $|b|_m$ then

$$\begin{aligned} a + b &\equiv a' + b' \pmod{m} \\ ab &\equiv a'b' \pmod{m} \end{aligned}$$

Indeed, there exist two integers x_1 and x_2 such that $a = a' + x_1 m$ and $b = b' + x_2 m$, from which we have $(a+b) = (a'+b') + (x_1+x_2)m$ and $(ab) = (a'b') + (a'x_2 + b'x_1 + x_1x_2m)m$, which proves the property (cf. (3.3)).

Example 7. *The tables of addition and multiplication modulo* 3 *are:*

$+$	0	1	2
0	0	1	2
1	1	2	0
2	2	0	1

$\times$	0	1	2
0	0	0	0
1	0	1	2
2	0	2	1

The tables of addition and multiplication modulo 4 *are:*

$+$	0	1	2	3
0	0	1	2	3
1	1	2	3	0
2	2	3	0	1
3	3	0	1	2

$\times$	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	0	2
3	0	3	2	1

We can easily show that $\mathbb{Z}_m$ is closed under modulo m addition, which is associative, has a identity element ($|0|_m$), an inverse for each class $|a|_m$ ($|-a|_m$) and is commutative. It follows that $\mathbb{Z}_m$ forms a commutative group with regard to addition. Furthermore, $\mathbb{Z}_m$ is closed under multiplication modulo m, which is also associative, has a neutral element ($|1|_m$) and is commutative. Therefore, $\mathbb{Z}_m$ is a commutative semi-group with regard to multiplication. Consequently, $\mathbb{Z}_m$ is a commutative ring.

Multiplication does not always admit an inverse. Indeed, if for example $m = 4$, $|2|_4$ does not have an inverse: none of the classes, $|1|_4$, $|2|_4$, $|3|_4$, when multiplied by $|2|_4$ results in $|1|_4$. However, if $m = 3$, we see that $|1|_3 \cdot |1|_3 = |1|_3$ whereas $|2|_3 \cdot |2|_3 = |1|_3$. Consequently each non-zero element has a multiplicative inverse and $\mathbb{Z}_3$ is a field.

We claim that a has a multiplicative inverse modulo m if and only if $\gcd(a, m) = 1$. Let us first show that a has a multiplicative inverse if $\gcd(a, m) = 1$. Indeed, we can explicitly construct the multiplicative inverse of a by using the extended Euclidian algorithm. We have

$$\begin{aligned} \alpha a + \mu m &= \gcd(a, m) = 1 \\ \alpha a &\equiv 1 \pmod{m} \\ \alpha &= a^{-1}. \end{aligned}$$

Assume now, conversely, that a has an inverse modulo m, *i.e.,*

$$\begin{aligned} a \cdot a^{-1} &\equiv 1 \pmod{m} \\ a \cdot a^{-1} + bm &= 1 \qquad \text{for some } b \in \mathbb{Z}. \end{aligned} \tag{3.4}$$

As $\gcd(a, m)$ is a divisor of both a and m, it also divides any linear combination of a and m. Therefore, according to (3.4), 1 should be divisible by $\gcd(a, m)$, implying that $\gcd(a, m) = 1$, which establishes the claim. When m is prime, which is to say that $\gcd(a, m) = 1$ for all a such that $0 < a < m$, all non-zero elements of $\mathbb{Z}_m$ have an inverse. So $\mathbb{Z}_m$ is a field if and only if m is prime. In this case, we call $\mathbb{Z}_m$ a *Galois field*, written $\mathcal{GF}(m)$.

A particularly easy operation to do is *exponentiation modulo* m, that is

$$x \equiv a^n \pmod{m}$$

Indeed, let's calculate for example the residue of 3^{12} modulo 7. We can successively reduce the base modulo 7 in the following way:

$$|3^{12}|_7 = (|3^2|_7)^6 = (|2|_7)^6 = |64|_7 = |1|_7,$$

and so

$$3^{12} \equiv 1 \pmod{7}.$$

However, the calculation of the *discrete logarithm* of a number x modulo m, that is determining x such that

$$a^x \equiv b \pmod{m}$$

is markedly more difficult and no efficient algorithm is known. For instance, to find $2^x \equiv 3 \pmod{13}$, we can successively try

$$\begin{aligned} 2^1 &\equiv 2 \pmod{13} \\ 2^2 &\equiv 4 \pmod{13} \\ 2^3 &\equiv 8 \pmod{13} \\ 2^4 &\equiv 3 \pmod{13} \\ 2^5 &\equiv 6 \pmod{13} \\ &\vdots \end{aligned}$$

from which we conclude that $x = 4$. This procedure becomes very inefficient for large values of m, and furthermore a congruence of this type does not always have a solution. As the exponential function is easy to compute, but its inverse, the logarithm is hard to compute, we say that this function is a *one-way function*. In the same way, a phone-book is a one-way function, because it is easy to look up someone's number given their name, but it is much more cumbersome to find the name of a person given only their phone number.

3.3.7 Euler's Totient Function

The *complete* set of modulo m residues is $\{0, 1, 2, \ldots, m-1\}$. The *reduced* set of modulo m residues is the set of modulo m residues which are coprime to m.

Example 8. *The complete set of modulo* 10 *residues is* $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ *but the reduced set of modulo* 10 *residues is* $\{1, 3, 7, 9\}$.

Euler's totient function (or *Euler phi function*) $\phi(m)$ is the number of elements of the reduced set of modulo m residues. For our example, we have $\phi(10) = 4$. Note that $\phi(1) = 1$.

If m is a prime number, each non-zero residue $r = 1, 2, \ldots, m-1$ is coprime to m. Consequently, $\phi(m) = m - 1$ in this case.

Furthermore, if m is prime, the reduced set of modulo m^2 residues is $\{1, 2, \ldots, m-1, m+1, \ldots, 2m-1, 2m+1, \ldots, m^2-1\}$ and thus contains $m^2 - 1 - (m-1) = m(m-1)$ elements, so $\phi(m^2) = m(m-1)$. In general, we can show that $\phi(m^n) = m^{n-1}(m-1)$ if m is prime.

The Euler function has the interesting property of preserving multiplication, that is to say

$$\phi(mn) = \phi(m)\phi(n)$$

if m and n are relatively prime. Consequently, if an integer m is decomposed into prime factors p_i:

$$m = \prod_i p_i^{e_i}$$

with $e_i > 0$, its Euler totient function will be

$$\phi(m) = \prod_i p_i^{e_i - 1}(p_i - 1) = m \prod_i (1 - \frac{1}{p_i}).$$

3.3.8 Euler's Theorem

Theorem 2. *If a and m are relatively prime,*

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

Let's prove this theorem. Let $\{r_1, r_2, \ldots, r_{\phi(m)}\}$ be the reduced set of modulo m residues. Multiply each of them by a and examine the set of residues $\left\{|ar_1|_m, |ar_2|_m, \ldots, |ar_{\phi(m)}|_m\right\}$.

As on the one hand a and m are relatively prime, and on the other hand r_i and m are also pairwise relatively prime for all $1 \leq i \leq \phi(m)$, ar_i and m are relatively prime. Consequently, $|ar_i|_m = |r_j|_m$ for a certain $1 \leq j \leq \phi(m)$ (see exercise 15). Also, if $i \neq k$, then $|r_i|_m \neq |r_k|_m$, and (see exercise 16)

$$|ar_i|_m = |a|_m \cdot |r_i|_m \neq |a|_m \cdot |r_k|_m = |ar_k|_m,$$

which shows that each residue of ar_i modulo m is a different member of the reduced set of modulo m residues, and so

$$\left\{|ar_1|_m, |ar_2|_m, \ldots, |ar_{\phi(m)}|_m\right\} = \left\{|r_1|_m, |r_2|_m, \ldots, |r_{\phi(m)}|_m\right\}.$$

From this, it follows that

$$\begin{aligned} \left|a^{\phi(m)} r_1 r_2 \ldots r_{\phi(m)}\right|_m &= |ar_1|_m \cdot |ar_2|_m \cdot \ldots \cdot |ar_{\phi(m)}|_m \\ &= |r_1|_m \cdot |r_2|_m \cdot \ldots \cdot |r_{\phi(m)}|_m \\ &= \left|r_1 r_2 \ldots r_{\phi(m)}\right|_m \end{aligned}$$

and so, as r_i and m are relatively prime for all $1 \leq i \leq \phi(m)$,

$$\left|a^{\phi(m)}\right|_m = |1|_m.$$

QED

3.3.9 Fermat's Theorem

Theorem 3. *A direct corollary of Euler's theorem is Fermat's theorem: if m is prime and if a is not divisible by m, then*

$$a^{m-1} \equiv 1 \pmod{m} \tag{3.5}$$

3.3.10 Calculating Modular Inverses

We have shown that a has an inverse modulo m, i.e., that there exists an element (a^{-1}) such that

$$a.a^{-1} \equiv 1 \pmod{m}, \tag{3.6}$$

if and only if a and m are relatively prime. There are three ways to compute this inverse:

1. Examine all the numbers $1, 2, \ldots, m-1$ until a number is found satisfying (3.6).
2. If the Euler phi function $\phi(m)$ is known, we compute

   $$a^{-1} \equiv a^{\phi(m)-1} \pmod{m}. \tag{3.7}$$

   Euler's theorem then guarantees that $a.a^{-1} \equiv 1 \pmod{m}$.
3. Starting from the Bézout's identity (3.1), we have that $\alpha a + \mu m = \gcd(a, m) = 1$. This shows that $a^{-1} = \alpha$ as $\alpha a \equiv 1 \pmod{m}$. We can use Euclid's extended algorithm to compute α.

3.3.11 Chinese Remainder Theorem

Theorem 4. *Let m_1 and m_2 be two relatively prime numbers. Then, for all integers r_1 and r_2, there exists an integer r such that any integer a which satisfies the congruences*

$$a \equiv r_1 \pmod{m_1} \tag{3.8}$$

$$a \equiv r_2 \pmod{m_2} \tag{3.9}$$

also satisfies

$$a \equiv r \pmod{m_1 m_2}. \tag{3.10}$$

To show this result, consider two integers u and v which satisfy Bézout's identity

$$um_1 + vm_2 = 1, \tag{3.11}$$

and let

$$r = r_2 u m_1 + r_1 v m_2. \tag{3.12}$$

Then, $r \equiv r_1 v m_2 \pmod{m_1}$, but Bézout's identity (3.11) implies that $vm_2 \equiv 1 \pmod{m_1}$, and thus $r \equiv r_1 \pmod{m_1}$. The same reasoning gives $r \equiv r_2 \pmod{m_2}$, and a satisfies the first two congruences (3.8) et (3.9).

If a is an integer which satisfies these congruences, we have $a - r \equiv 0 \pmod{m_1}$ and $a - r \equiv 0 \pmod{m_2}$. Thus, m_1 and m_2 divide $a - r$. As these two numbers are coprime, their product $m_1 m_2$ also divides $a - r$, and a satisfies the third congruence (3.10). QED.

This theorem can be extended by recurrence to any number of congruences.

3.4 Asymmetric Encryption Algorithms

We now have the tools to discuss public key cryptography.

3.4.1 Conditions for Setting up a Public Key Cryptosystem

Let us remind ourselves that an encryption algorithm E, parameterized by a key K, transforms the plaintext P into the ciphertext $C = E_K(P)$. The deciphering algorithm D, parametrized by a key k, does the inverse transformation $P = D_k(C)$.

In Section 3.2, we had only considered symmetric ciphers, for which the same key was used both for encryption and decryption: $K = k$. Such a cryptosystem therefore requires that the secret key be previously communicated between the sender and receiver, which is a delicate operation, especially if the key must be broadcast to a large number of users, as is the case for computer networks. If the secret key falls in the hands of the enemy, the whole system is compromised.

A public key cryptosystem gets rid of this problem by allowing different keys for encryption and decryption: $K \neq k$. The key K is public whereas the key k is secret. In this way, the encryption key K can be transmitted along a non-secure channel (cf. Fig 3.3), as the enemy knows it.

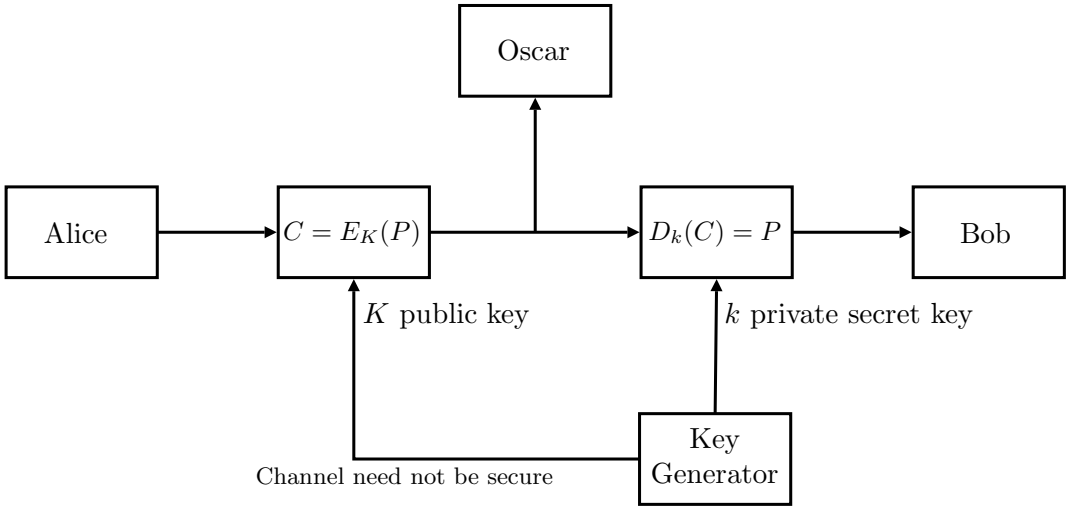

Figure 3.3: Asymmetric Cipher.

In order that a public key cryptosystem works correctly, several conditions must be met:

1. The encryption $E_K(P)$ of plaintext message P is fast and easy to compute.
2. The decryption $D_k(C)$ of ciphertext C is fast and easy to compute, provided the receiver knows the key k.
3. However, computing $E_K^{-1}(C)$ is extremely difficult and impossible to achieve in practice, within a reasonable timeframe.
4. Lastly, as in any cryptosystem, the decryption algorithm must be the inverse of the encryption algorithm: $D_k(E_K(P)) = P$.

As we can see, E_K is a one-way function. Furthermore, because knowing the key k makes calculating the inverse easy, it is no longer quite a one-way function and becomes a *trap-door function*.

So a user who wishes to send and receive ciphertexts uses two keys, one of them (k) he keeps secrete and the other (K), he publishes in a directory where all the other users can look it up. Key distribution is thus greatly facilitated.

3.4.2 The Rivest-Shamir-Adleman Algorithm (RSA)

In a RSA cryptosystem, plaintext messages P, ciphertexts C and keys K, k are coded as integers modulo m and are all included between 1 and $m-1$. Consequently, P, C, K and $k \in \mathbb{Z}_m$. The module m is made public. The encryption and decryption algorithms are respectively

$$E_K(P) = C \equiv P^K \pmod{m} \tag{3.13}$$

with $1 \leq C \leq m-1$ and

$$D_k(C) = P \equiv C^k \pmod{m} \tag{3.14}$$

with $1 \leq P \leq m-1$, where the keys are chosen so that

$$\begin{aligned} \text{pgcd}(K, \phi(m)) &= 1 && (3.15) \\ Kk &\equiv 1 \pmod{\phi(m)} && (3.16) \end{aligned}$$

with $1 \leq K, k \leq \phi(m) - 1$. Note that this last congruence (3.16) indicates that k is the inverse of K modulo $\phi(m)$ and that this inverse always exists since K and $\phi(m)$ are relatively prime (3.15).

Example 9. *Consider the following trivial example. Each letter of the plaintext is encoded by its alphabetic position and we take $m = 33$, from which $\phi(m) = \phi(11)\phi(3) = 20$. Let's choose $K = 7$, which is relatively prime to $\phi(m)$. We can then calculate $k = 3$ as $7 \cdot 3 \equiv 1 \pmod{20}$.*

Consequently, the encryption and decryption of the word "goodbye" become:

	plaintext P	*ciphertext* $C = \|P^7\|_{33}$	*decryption* $\|C^3\|_{33}$	
G	07	28	07	G
O	15	27	15	O
O	15	27	15	O
D	04	16	04	D
B	02	29	02	B
Y	25	31	25	Y
E	05	14	05	E

We must now check that this algorithm satisfies the conditions laid out in Section 3.4.1 which define a public key cryptosystem.

1. First of all, **encryption** is an exponentiation operation, which is **fast**, as we saw in the example in Section 3.3.6, and is furthermore **easy** to implement in hardware.

2. **Decryption** is also an exponentiation operation, which is also **fast and easy so long as k is known**.

3. In order to satisfy the third condition, it must be **very difficult to discover C and k if k is not known**.

Let us first suppose that m is prime. In that case, $\phi(m) = m - 1$ is known since m is public and the congruence (3.16) becomes

$$Kk \equiv 1 \pmod{m-1}.$$

Consequently, k is is equal to K^{-1} modulo $m - 1$, and can be computed using methods seen in Section 3.3.10. We know that this computation must be easy in order to allow the user to compute k when setting up his cryptosystem. This would allow the cryptanalyst to compromise the system, which is unacceptable. Clearly, we must choose m such that $\phi(m)$ be difficult to compute.

We shall therefore choose an m which is not prime,

$$m = pq \tag{3.17}$$

where p and q are prime numbers, called *initial conditions* of the algorithm. Now, $\phi(m) = (p-1)(q-1)$ can only be known if we know the two prime factors p and q. Consequently, the regular user will publish K and m but neither k nor p nor q which he will therefore be the only one to know. Only he will be able to calculate k relatively easily knowing K using one of the algorithms seen in Section 3.3.10, because he knows $\phi(m)$. The cryptanalyst, however, not having access to $\phi(m)$, will find it much more difficult to compute k. We will see in the following section that he will be unable to compromise the system within a reasonable timeframe, which shows that the third condition is satisfied.

4. Lastly, we must check that **the functions E and D are inverses**, that is to say $D_k(E_K(P)) = P$ or in other terms that

$$(P^K)^k \equiv P \pmod{m}. \tag{3.18}$$

As (3.16) is satisfied, there must exist an integer x such that $Kk = 1 + x\phi(m)$ and thus

$$P^{Kk} = P^{1+x\phi(m)} = P \cdot (P^{\phi(m)})^x.$$

If P et m are relatively prime, Euler's theorem shows that $P^{\phi(m)} \equiv 1 \pmod{m}$. So

$$|P \cdot (P^{\phi(m)})^x|_m = |P|_m \cdot (|P^{\phi(m)}|_m)^x = |P|_m \cdot |1^x|_m = |P|_m \cdot |1|_m = |P|_m,$$

and (3.18) is satisfied.

Suppose that P and m are not relatively prime, which is quite rare if m is large. As $m = pq$ with p and q prime, we must have either $P = ip$ ou $P = jq$. Suppose without loss of generality that $P = ip$. In this case, $|P|_p = |i|_p.|p|_p = |0|_p$ and so

$$P^{x\phi(m)} \equiv 0 \pmod{p}. \tag{3.19}$$

On the other had, as q is prime and does not divide P, Fermat's theorem results in

$$P^{q-1} \equiv 1 \pmod{q},$$

and therefore that

$$P^{x(p-1)(q-1)} = P^{x\phi(m)} \equiv 1 \pmod{q}. \tag{3.20}$$

Consequently, the congruences (3.19) (3.20) together with the Chinese remainder theorem imply that there exists and integer r such that

$$P^{x\phi(m)} \equiv r \pmod{pq},$$

and so

$$P^{Kk} = P^{1+x\phi(m)} \equiv Pr \pmod{pq}. \tag{3.21}$$

If u and v are two integers satisfying Bézout's identity $up + vq = 1$, the relation (3.12) allows us to determine this integer r: $r = up \cdot 1 + vq \cdot 0 = up$, or even $r = 1 - vq$. From that we have

$$|Pr|_{pq} = |P|_{pq} \cdot |1 - vq|_{pq} = |P|_{pq} \cdot (|1|_{pq} - |vq|_{pq}) = |P|_{pq} - |Pvq|_{pq} = |P|_{pq} - |ivpq|_{pq} = |P|_{pq}$$

and (3.21) becomes

$$P^{Kk} \equiv P \pmod{pq},$$

which shows that the last condition set in the previous section is fulfilled.

3.4.3 Security of the RSA Algorithm

How would a cryptanalyst go about breaking an RSA cryptosystem? There are three possible angles of attack.

1. He can first try to determine the *initial conditions* p and q. He can then easily compute $\phi(m) = (p-1)(q-1)$ and then find k from (3.16). But if m is very large, **factorisation** of m into two prime number is very difficult. One method would be to try all prime numbers up

to $\sqrt{m}$. Even the most efficient algorithms know to date are unable to factor large integers in reasonable time. Rivest, Shamir and Adleman estimated that it would take 4 million years to factorise a 200 digit number m with a 1Ghz processor.

2. He can try to solve

$$C \equiv P^K \pmod{m}$$

with C, K and m known in order to find the plaintext P (the key k will still remain unknown). He is then confronted with the problem of computing a **discrete logarithm**, which is very difficult, as seen in Section 3.3.6.

3. A third possible attack is **successive encryption of the ciphertext**. Indeed, since the cryptanalyst knows K, m and C, he can encrypt the ciphertext C himself giving a new ciphertext C_1:

$$C_1 = E_K(C) \equiv C^K \pmod{m}.$$

He can repeat the same operation on C_1, giving C_2 and so on for any i:

$$C_{i+1} = E_K(C_i) \equiv C_i^K \pmod{m}.$$

As $1 \leq C \leq m-1$, there are a finite number of C_i and it is possible that for a certain $i = n$, $C_n = C$. In that case,

$$C_n = C = E_K(C_{n-1}) = E_K(P) \equiv P^K \pmod{m}$$

and so the ciphertext C_{n-1} must be the same as the original plaintext P. This integer n, called the *order* of $E(\cdot)$, must therefore be as large as possible so that such an attack be impossible to realise in a relatively short time. Rivest showed that if the integers $p-1$ and $q-1$ had high prime factors (for instance $p-1 = 2p'$ and $q-1 = 2q'$, where p' and q' are prime) the probability of such an attack succeeding was near zero.

Consequently, cryptanalysis cannot, in practice, compromise an RSA cryptosystem.

3.4.4 Notes on the Choice of Parameters for an RSA Cryptosystem

While the cryptanalyst is fortunately presented with unsurmountable problems this does not mean that the cryptographer's choices are trivial. Two problems must be considered:

1. In order to resist the third type of attack mentioned in the previous Section, $p-1$ and $q-1$ must have large prime factors. Even once p' and q' are chosen, we must check that p and q are prime. Typically, p and q are 100 digit numbers, so that m be a 200 digit number. It is much easier to test if a 100 digit number is prime than to factor a 200 digit number (which the cryptanalyst's first attack attempts to do), but this nevertheless takes time.

2. With the encryption given in Section 3.4.2, we notice that the numbers 10 and 21 of the plaintext are encrypted respectively to 10 and 21. Indeed, $10 \equiv 10^7 \pmod{33}$ and $21 \equiv 21^7 \pmod{33}$, so the ciphertext and plaintext are identical for these values; this is obviously

an advantage for the cryptanalyst. The RSA algorithm does not hide all plaintexts P and sometimes $C = P$, which means

$$P^K \equiv P \pmod{m}.$$

We can show that the number σ of messages $1 \leq P \leq m-1$ that the cryptosystem cannot hide is equal to

$$\sigma = (1 + \gcd(K-1, p-1))(1 + \gcd(K-1, q-1)) - 1.$$

The key K must thus be chosen in order to minimise σ. If the key is chosen without regard for this, more than half of the plaintext may stay disclosed!

Lastly, let us note that the RSA algorithm can also be used for other tasks than information protection, such as *authentification*.

3.5 Bibliographical References

D. R. Stinson, *Cryptography: Theory and Practice*. CRC Press, 1995.

Enigma Sibulator, Bletchley Park Website, `http://www.bletchleypark.org.uk`

C. E. Shannon, *Prediction and Entropy of Printed English*. Bell System Technical Journal, Vol 30 (1951), pp.50-64 Reprinted in D. Slepian, editor, Key Papers in the Development of Information Theory, IEEE Press, NY, 1974.

3.6 Exercises

Exercise 1. Break the following ciphertexts which have been encrypted using a monolalphabetic substitution which rotates the letters of the alphabet k positions. (NB The plaintext is in French.)

a) TU JEKI BUI FUKFBUI TU BQ WQKBU BUI RUBWUI IEDJ BUI FBKI RHQLUI

b) JK ZUAY RKY GAZXOINOKTY RK VRAY IKRKHXK KYZ IGROLUXTOKT

The first sentence (a) is a French translation of a sentence said by Julius Caesar taken from "La Guerre des Gaules". The second (b) is a particular point of view.

Exercise 2. For the English language, what is the unicity distance of a monoalphabetic cipher that shifts the letters by k positions? What is the minimal length of the ciphertext that allows a cryptanalyst to break this cipher? (He knows the encryption method, but does not know k.) Show that if the length of the ciphertext known to the cryptanalyst is smaller than the unicity distance, then the cryptanalyst is not able to reconstruct the corresponding plaintext.

Exercise 3. Repeat exercise 2 for the Caesar cipher with $k = 3$

Exercise 4. The plaintext "thisisasample" is encrypted as "KVCFLTADLDDFR".

a) Was the cipher a monoalphabetic or polyalphabetic cipher (Vigneère cipher)?

b) Which key was used?

Exercise 5. The alphabet of a source is make up of five symbols which turn up independently from each other and each have the same probability $p_i = 0.2$ $(1 \leq i \leq 5)$ of being emitted. These symbols are encrypted by monoalphabetic substitution. Does this method seem sufficient so resist the attack of a cryptanalyst who only has the ciphertext and not the plaintext (ciphertext-only attack)? Why?

Exercise 6. Decrypt the ciphertext "EOBABNOKTOTEORTX" obtained by transposition with the key SURF.

Exercise 7. Calculate x such that

a) $x \equiv 4^8 \pmod{15}$

b) $x \equiv 1234^{567} \pmod{99}$

c) $x \equiv 2^{123}3^{456}5^{789} \pmod{4}$.

Exercise 8. Calculate Euler' totient function $\phi(m)$ fir $m = 29$, 30, 31 and 32.

Exercise 9. *Prove the following properties.*

- $\gcd(a, 1) = \gcd(a - a \cdot 1, 1) = \gcd(0, 1) = 1$,
- $\gcd(a, p) = 1$ *with* $a \neq 0$ *and* p *prime,*
- $\gcd(a, p) = p$ *with* $a = 0$ *and* p *prime.*

Exercise 10. Calculate the GCD of

a) 234 and 325

b) 97 and 109.

Exercise 11. Find two integers u and v which satisfy Bézout's Identity

$$234u + 325v = \text{pgcd}(234, 325).$$

Exercise 12. If they exist, calculate the following inverses:

a) $x \equiv 3^{-1} \pmod{10}$

b) $x \equiv 3^{-1} \pmod{15}$

c) $x \equiv 3^{-1} \pmod{23}$.

Exercise 13. Find an integer x which satisfies the congruences

$$\begin{aligned} x &\equiv 7 \pmod{11} \\ x &\equiv 6 \pmod{13}. \end{aligned}$$

Exercise 14. Show that the relation (3.2) is symmetrical and transitive.

Exercise 15. If $a \equiv b \pmod{m}$ and b and m are relatively prime, show that a et m are also relatively prime.

Exercise 16. a) If $a \equiv a' \pmod{m}$, show that for any integer t

$$at \equiv a't \pmod{m}.$$

b) If $ad \equiv a'd \pmod{m}$ and d and m are relatively prime, show that

$$a \equiv a' \pmod{m}.$$

Does this property still hold if d and m are not relatively prime?

Exercise 17. Must a public key cryptosystem be able to resist a ciphertext only attack? a known plaintext attack? a chosen plaintext attack?

Exercise 18. The initial conditions of an RSA cryptosystem are $p = 23$ et $q = 47$, whereas the public key is $K = 13$. What is the secret key?

Exercise 19. You are a cryptanalyst faced with an RSA cryptosystem where $m = 187$ and $K = 7$.

a) You have intercepted the ciphertext $C = 183$, which was sent by the regular user. Break this ciphertext without trying to find the secret key.

b) Compromise the whole system by working out the secret key.

Exercise 20. The initial conditions of an RSA cryptosystem are $p = 97$ and $q = 109$. Among the following public keys, which ones can effectively be chosen?

a) $K = 123$

b) $K = 865$

c) $K = 169$

Exercise 21. A cryptanalyst knows that, for $m = 10001$, $\phi(m) = 9792$. What are the initial conditions p and q ? (Hint: we can write $\phi(m)$ as $\phi(m) = m - p - q + 1$.)

Exercise 22. As explained in Section section 3.4.3, during the design phase of an RSA cryptosystem, it is useful to have an algorithm which quickly tests whether a given integer p is prime or not.

Wilson's theorem, which explains that

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

if and only if p is prime gives us such an test: the function

$$f(p) = \sin(\pi \, \frac{(p-1)! + 1}{p})$$

is zero if and only if p is prime. Is this test usable in practice?

Exercise 23. Instead of encrypting the plaintext with the public key and decrypting it with the secret key, we perform the opposite operation: the *encryption* key is the secret key k whereas the *decryption* key is the public key K. What could this system be used for?