

Number Theory – cryptosystems

8.1 Review

Let’s now look at expressing Property 1, Euclid’s Lemma and Fermat’s Little Theorem in terms of congruences.

Property 1. *If $b \equiv 0 \pmod{a}$ and $c \equiv 0 \pmod{a}$ then $bm + cn \equiv 0 \pmod{a}$, for all $m, n \in \mathbb{Z}$.*

Lemma 8.1.1 (Euclid’s Lemma). *If p is prime and $ab \equiv 0 \pmod{p}$ then*

$$a \equiv 0 \pmod{p} \quad \text{or} \quad b \equiv 0 \pmod{p}.$$

Theorem 8.1.2 (Fermat’s Little Theorem). *If p is prime and $a \not\equiv 0 \pmod{p}$ then $a^{p-1} \equiv 1 \pmod{p}$.*

Also, recall that two integers a, b are *coprime* if their greatest common divisor (a, b) is 1. We also saw the following lemma previously as two separate results: part (i) is Lemma 7.2.11, and (ii) is Corollary 7.2.12; in fact, (ii) generalises (i).

Lemma 8.1.3. (i) *If $ac \equiv bc \pmod{m}$ and $(c, m) = 1$ then $a \equiv b \pmod{m}$.*

(ii) *If $ac \equiv bc \pmod{m}$ and $(c, m) = d$ then $a \equiv b \pmod{m/d}$.*

8.2 Cryptosystems

A *code* is an algorithm used to reformulate a message in a convenient form for transmission, e.g. the *Morse code* *encodes* a message as dots and dashes, convenient for transmission along a wire. It was designed before it was discovered how to transmit voice messages. When one *encodes* a message, one is not trying to keep the message *secret*.


A *cryptosystem* is an algorithm used to *encrypt* a message to keep it *secret*. To describe some of these it will be useful to start with a numerical encoding of the alphabet and the blank space between words:

letter	space	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
encoding	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26

One of the simplest cryptosystems is the *Caesar cipher* which replaces each letter (with encoding u) of a message by the letter with encoding v , where:

$$v \equiv au + b \pmod{m}$$

where a, b, m are fixed integers such that a and m are coprime and m is at least as large as the number of letters in your alphabet. For our alphabet above we would need m to be at least 27.

 The cipher is named for Julius Caesar who used such a cryptosystem with $a = 1$ and $b = 3$. If we choose $m = 27$ with Julius Caesar’s choice for a and b encrypting a message amounts to a *cyclic shift* of each letter right by 3 letters. This simple example is easy to break ... by determining the encryption of just one letter of the message, by trial-and-error (there are only 27 possibilities to check ... and by picking on the most frequently occurring letter of the encrypted message we might try ‘e’ first, etc.).

Another simple cryptosystem is the *one-time pad*. The way this works is that both the receiver and sender have a long sequence of random numbers, (b_1, b_2, \dots) . If a message with the simple numerical encoding of the letters is:

$$u_1, u_2, \dots, u_i, \dots$$

then the i^{th} letter of the encoded message is encrypted as

$$u_i + b_i \pmod{m},$$

where m may be 27 if spaces are encoded or 26 if they are not encoded. Each sequence of random numbers is used just *once* which makes the cryptosystem *unbreakable*. The method is however extremely cumbersome, because both sender and receiver must keep a very long sequence of numbers.



A one-time pad is used for the hot-line between Washington and Moscow.

For frequent computer-based communication among several parties it is desirable to have a cryptosystem with neither of the faults of the above systems, i.e.

- (i) the *encryption* and *decryption* algorithms are easy to compute and reusable; and
- (ii) each person's *decryption* algorithm cannot be obtained from his/her *encryption* algorithm in any reasonable amount of time.

The second property means that the *encryption* algorithm can in fact be made public, and so such a system, together with the following property, is called a *public-key system*.

- (iii) For a *message* a , *encryption* algorithm E and *decryption* algorithm D both

$$D(E(a)) = a \quad \text{and} \quad E(D(a)) = a.$$

8.2.1 The RSA Cryptosystem

The RSA system is an example of a *public-key system* that was developed in 1977 by Rivest, Shamir and Adleman. It is based on the following result that follows from Fermat's Little Theorem.

Theorem 8.2.1.1 (RSA Theorem). *Let p, q be distinct primes;*

$$\begin{aligned} &\text{let } n = pq, \\ &\text{let } k = (p - 1)(q - 1), \\ &\text{choose } d \text{ coprime to } k, \text{ and} \\ &\text{choose } e \text{ such that } de \equiv 1 \pmod{k}. \end{aligned}$$

Then $a^{ed} \equiv a \pmod{n}$ for any integer a .

Proof. Firstly, the hypotheses of the theorem can be satisfied, since, if one chooses $d = k - 1$ then $(d, k) = 1$ and so, the Euclidean Algorithm guarantees a solution of

$$dx + ky = 1,$$

whence, for $e = x$ we have

$$de \equiv 1 \pmod{k}.$$

By Fermat's Little Theorem, if $a \not\equiv 0 \pmod{p}$ then

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

So, for any nonnegative integer ℓ ,

$$a^{\ell(p-1)+1} \equiv a \pmod{p},$$

and this is also true if $a \equiv 0 \pmod{p}$. Since $de \equiv 1 \pmod{k}$, where $k = (p-1)(q-1)$, we have $ed = de = \ell(p-1) + 1$ for some integer ℓ . So

$$a^{ed} \equiv a \pmod{p},$$

and hence

$$a^{ed} \equiv a + m_1q \pmod{pq}, \tag{8.2.1.1}$$

for some integer m_1 . Similarly,

$$a^{ed} \equiv a \pmod{q},$$

and hence

$$a^{ed} \equiv a + m_2p \pmod{pq}, \tag{8.2.1.2}$$


for some integer m_2 . Since p and q are distinct primes, it follows from (8.2.1.1) and (8.2.1.2) that

$$a^{ed} \equiv a \pmod{pq}. \quad \square$$

Here is an example to show how we use this result to come up with a *cryptosystem*.

Example 8.2.1.2. Suppose our message is 'GO WEST'. Then we perform the following steps.


1. Encode the letters numerically, e.g. by the encoding given in the table on page 67. This gives: 07150023051920. Call this number a .
2. We need p, q such that $n = pq > a$.
3. To encrypt the message, compute: $E(a) = a^e \pmod{n}$.
4. To decrypt the message, the receiver computes: $D(E(a)) = (a^e \pmod{n})^d \pmod{n}$ which by the RSA Theorem is congruent to $a \pmod{n}$, and since $n > a$ we know the message was a .

 In computing, the *binary operators* `div` and `mod` are defined as follows: if for integers a, m , with m positive, the *quotient* and *remainder* when a is divided by m are q and r respectively, i.e. $a = mq + r$, then

$$\begin{aligned} a \operatorname{div} m &= q, \\ a \operatorname{mod} m &= r. \end{aligned}$$


In particular, if $r = a \operatorname{mod} m$ then $r \equiv a \pmod{m}$ and $0 \leq r < m$. Take careful notice of how the *computing* syntax for `mod` differs from the usual *mathematics* usage. We defined these operations in Definition 7.2.3.

Observe that in our example the message a was a rather large number and that n needed to be even larger. In practice, p and q are large primes (in 2023, RSA primes are 600–1200 digits each). Observe that choosing appropriate p, q, d determines n, k, e . The numbers e and n for the cryptosystem are publicly announced. The system is secure since determining the decryption algorithm is at least as difficult as factoring n .

 With the technology of 1990, the expected time to factor a 200-digit number that is the product of two equal length primes, was approximately 4 million years. Ironically, 33 years later, it is estimated with 2023 technology, the same task would take 75 years.

Observe, also that Property (iii) of a *public-key system* is satisfied since

$$(a^d)^e = (a^e)^d \equiv a \pmod{pq}.$$

 The significance of this is that a sender may use their *decryption* algorithm to encrypt a signature s as $s^d \pmod{n}$. Then, the receiver (and in fact, anybody) can use the publicly available *encryption* algorithm to decrypt it again, and so verify the source of the message.

Very long messages may still give a number larger than the simple encoding a , in which case one needs to break up the message into modules and encrypt each module separately. Here is one of Rivest, Shamir and Adleman's own examples.

Example 8.2.1.3. Take the message: “IT’S ALL GREEK TO ME” and suppose $n = 2773$, $d = 157$ and $e = 17$. Since we can only encode numbers less than 2773, we choose blocks of length 2.

1. Numerical encoding of the blocks gives (where # denotes a blank space and other punctuation is ignored):

I	T	S	#	A	L	L	#	G	R
09	20	19	00	01	12	12	00	07	18
E	E	K	#	T	O	#	M	E	#
05	05	11	00	20	15	00	13	05	00

2. Observe each block is encoded with a number less than 2773.
3. Encrypting the first block we have: $920^e = 920^{17} \equiv 948 \pmod{2773}$. Encrypting all the blocks we get the following coded form of the message:

09	48	23	42	10	84	14	44	26	63
23	93	07	78	07	74	02	19	16	55

4. The receiver would now decrypt the message by applying the decryption algorithm to each block, e.g. for the first block

$$948^d = 948^{157} \equiv 920 \pmod{2773}$$

which is the numerical encoding for the first original block: ‘IT’.

Exercise Set 8.

1. Using the *Caesar cipher*, with a, b, m as defined in the dangerous bend on page 67 of the notes, encrypt: CRYPTOLOGY.

Answer. With $a = 1, b = 3$ and $m = 27$, the *Caesar cipher* amounts to being a cyclic shift of each letter by three letters. Hence CRYPTOLOGY is encrypted as:

FUASWRORJA

- *2. Decrypt the following message. Spaces are also encoded. There is one space in the encrypted output.

RUOELTWK EINHXFEQHZEYTDJPEHVONERUOEBGCAEMHS

3. Find the values of p, q, k for Example 8.2.1.3 ($n = 2773, d = 157$ and $e = 17$).
 4. Use $e = 3$ and $n = 2773$ to encrypt the following message using the RSA cryptosystem:

CODING IS EASY

Use 2-letter blocks and don't omit spaces.

- *5. Decrypt the following message. Spaces are also encoded. (It just so happens that no spaces appear after the encryption.)

BKDAKUNFKDWTDBJKNWKFNANTTNLKWNTKKBKIDS
 CKMCCUKYCMFCTJDYKDUJKBKHNCKFNJDY

Note that a *Caesar cipher* has been used (i.e. $\langle \text{SPACE} \rangle, A, \dots, Z$ are encoded as $00, 01, \dots, 26$ (as per the table on page 67), the *Caesar cipher* algorithm

$$v \equiv au + b \pmod{27}$$

has been applied for each letter u of the message for some a, b (which you essentially have to find), and the encoded letter v has been changed back to a letter using the table on page 67 again.)

Note: Letters and spaces occurring in English text, arranged approximately in order of highest frequency to lowest frequency are

$\langle \text{SPACE} \rangle, E, T, A, I, O, N, S, H, R, D, L, U, \dots$

Also, use the fact that inter-word spaces occur on average every 4–5 letters and use what you know about the possibilities of letters in short words of 1, 2 or 3 letters.

If this problem seems too hard, try doing it without using the fact that a *Caesar cipher* has been used.

Hint. Since you want to *decrypt* you really want to express u in terms of v , i.e. you really want to find a c, d such that

$$u \equiv cv + d \pmod{27}.$$

6. Use $e = 3$ and $n = 2773$ to encrypt the following message using the RSA cryptosystem:

THE HUNS ARE COMING

Use 2-letter blocks and don't omit spaces.

- *7. Find the decryption algorithm for the previous problem.