2.1 INTEGER ARITHMETIC

In **integer arithmetic,** we use a set and a few operations. You are familiar with this set and the corresponding operations, but they are reviewed here to create a background for modular arithmetic.

Set of Integers

The **set of integers,** denoted by **Z,** contains all integral numbers (with no fraction) from negative infinity to positive infinity (Figure 2.1).

Figure 2.1 *The set of integers*

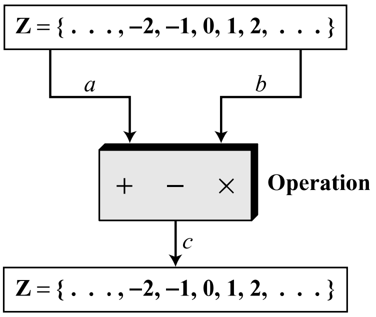
Z = { . . . , –2, –1, 0, 1, 2, . . . }

Binary Operations

In cryptography, we are interested in three binary operations applied to the set of integers. A **binary operation** takes two inputs and creates one output. Three common binary operations defined for integers are *addition, subtraction,* and *multiplication*. Each of these operations takes two inputs (a and b) and creates one output (c) as shown in Figure 2.2. The two inputs come from the set of integers; the output goes into the set of integers.

Note that *division* does not fit in this category because, as we will see shortly, it produces two outputs instead of one.

Figure 2.2 *Three binary operations for the set of integers*


Z = { . . . , –2, –1, 0, 1, 2, . . . }
a
b
+ – ×
Operation
c
Z = { . . . , –2, –1, 0, 1, 2, . . . }

Example 2.1

The following shows the results of the three binary operations on two integers. Because each input can be either positive or negative, we can have four cases for each operation.

Add:	5 + 9 = 14	(–5) + 9 = 4	5 + (–9) = –4	(–5) + (–9) = –14
Subtract:	5 – 9 = –4	(–5) – 9 = –14	5 – (–9) = 14	(–5) – (–9) = +4
Multiply:	5 × 9 = 45	(–5) × 9 = –45	5 × (–9) = –45	(–5) × (–9) = 45

Integer Division

In integer arithmetic, if we divide *a* by *n*, we can get *q* and *r*. The relationship between these four integers can be shown as

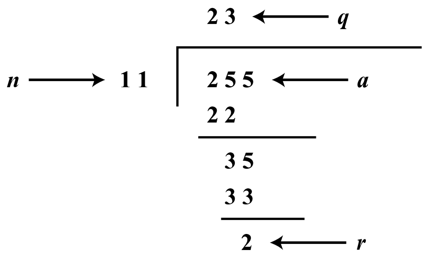
$$a = q \times n + r$$

In this relation, *a* is called the *dividend; q*, the *quotient; n*, the *divisor;* and *r*, the *remainder*. Note that this is not an operation, because the result of dividing *a* by *n* is two integers, *q* and *r*. We can call it *division relation*.

Example 2.2

Assume that $a = 255$ and $n = 11$. We can find $q = 23$ and $r = 2$ using the division algorithm we have learned in arithmetic as shown in Figure 2.3.

Figure 2.3 *Example 2.2, finding the quotient and the remainder*

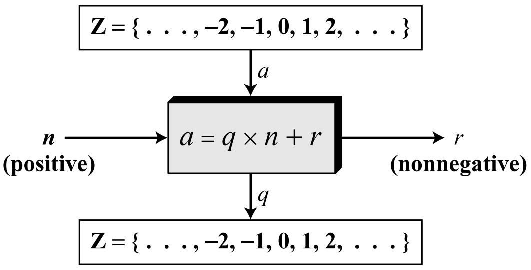


Most computer languages can find the quotient and the remainder using language-specific operators. For example, in the C language, the operator / can find the quotient and the operator % can find the remainder.

Two Restrictions

When we use the above division relationship in cryptography, we impose two restrictions. First, we require that the divisor be a positive integer ($n > 0$). Second, we require that the remainder be a nonnegative integer ($r \geq 0$). Figure 2.4 shows this relationship with the two above-mentioned restrictions.

Figure 2.4 *Division algorithm for integers*


Z = { . . . , –2, –1, 0, 1, 2, . . . }
a
n
(positive)
a = q × n + r
r
(nonnegative)
q
Z = { . . . , –2, –1, 0, 1, 2, . . . }

Example 2.3

When we use a computer or a calculator, *r* and *q* are negative when *a* is negative. How can we apply the restriction that *r* needs to be positive? The solution is simple, we decrement the value of *q* by 1 and we add the value of *n* to *r* to make it positive.

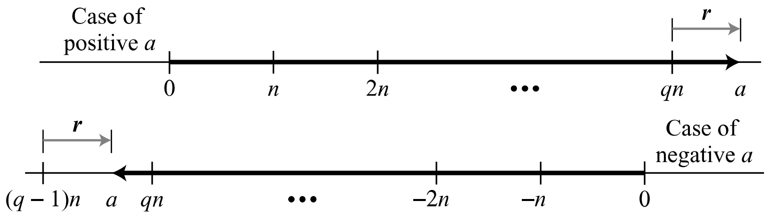
$$-255 = (\mathbf{-23} \times 11) + (\mathbf{-2}) \qquad \leftrightarrow \qquad -255 = (\mathbf{-24} \times 11) + \mathbf{9}$$

We have decremented −23 to become −24 and added 11 to −2 to make it 9. The above relation is still valid.

The Graph of the Relation

We can show the above relation with the two restrictions on *n* and *r* using two graphs in Figure 2.5. The first one shows the case when *a* is positive; the second when *a* is negative.

Figure 2.5 *Graph of division algorithm*



Starting from zero, the graph shows how we can reach the point representing the integer *a* on the line. In case of a positive *a*, we need to move $q \times n$ units to the right and then move extra *r* units in the same direction. In case of a negative *a*, we need to move $(q - 1) \times n$ units to the left (*q* is negative in this case) and then move *r* units in the opposite direction. In both cases the value of *r* is positive.

Divisibility

Let us briefly discuss **divisibility,** a topic we often encounter in cryptography. If *a* is not zero and we let $r = 0$ in the division relation, we get

$$\boldsymbol{a = q \times n}$$

We then say that *n* divides *a* (or *n* is a divisor of *a*). We can also say that *a* is divisible by *n*. When we are not interested in the value of *q*, we can write the above relationship as $\boldsymbol{a|n}$. If the remainder is not zero, then *n* does not divide *a* and we can write the relationship as $\boldsymbol{a \nmid n}$.

Example 2.4

a. The integer 4 divides the integer 32 because $\mathbf{32} = 8 \times \mathbf{4}$. We show this as $4|32$.
b. The number 8 does not divide the number 42 because $\mathbf{42} = 5 \times \mathbf{8} + 2$. There is a remainder, the number 2, in the equation. We show this as $8 \nmid 42$.

Example 2.5

a. We have $13|78$, $7|98$, $-6|24$, $4|44$, and $11|(-33)$.

b. We have $13 \nmid 27$, $7 \nmid 50$, $-6 \nmid 23$, $4 \nmid 41$, and $11 \nmid (-32)$.

Properties

Following are several properties of divisibility. The interested reader can check Appendix Q for proofs.

Property 1: if $a|1$, then $a = \pm 1$.
Property 2: if $a|b$ and $b|a$, then $a = \pm b$.
Property 3: if $a|b$ and $b|c$, then $a|c$.
Property 4: if $a|b$ and $a|c$, then $a|(m \times b + n \times c)$, where m and n are arbitrary integers.

Example 2.6

a. Since $3|15$ and $15|45$, according to the third property, $3|45$.

b. Since $3|15$ and $3|9$, according to the fourth property, $3|(15 \times 2 + 9 \times 4)$, which means $3|66$.

All Divisors

A positive integer can have more than one divisor. For example, the integer 32 has six divisors: 1, 2, 4, 8, 16, and 32. We can mention two interesting facts about divisors of positive integers:

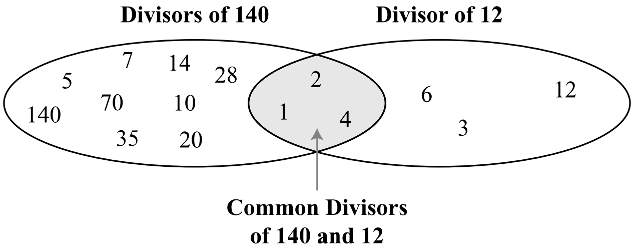
Fact 1: The integer 1 has only one divisor, itself.

Fact 2: Any positive integer has at least two divisors, 1 and itself (but it can have more).

Greatest Common Divisor

One integer often needed in cryptography is the **greatest common divisor** of two positive integers. Two positive integers may have many common divisors, but only one greatest common divisor. For example, the common divisors of 12 and 140 are 1, 2, and 4. However, the greatest common divisor is 4. See Figure 2.6.

Figure 2.6 *Common divisors of two integers*



The greatest common divisor of two positive integers is the largest integer that can divide both integers.

Euclidean Algorithm

Finding the greatest common divisor (gcd) of two positive integers by listing all common divisors is not practical when the two integers are large. Fortunately, more than 2000 years ago a mathematician named Euclid developed an algorithm that can find the greatest common divisor of two positive integers. The **Euclidean algorithm** is based on the following two facts (see Appendix Q for the proof):

Fact 1: gcd $(a, 0) = a$

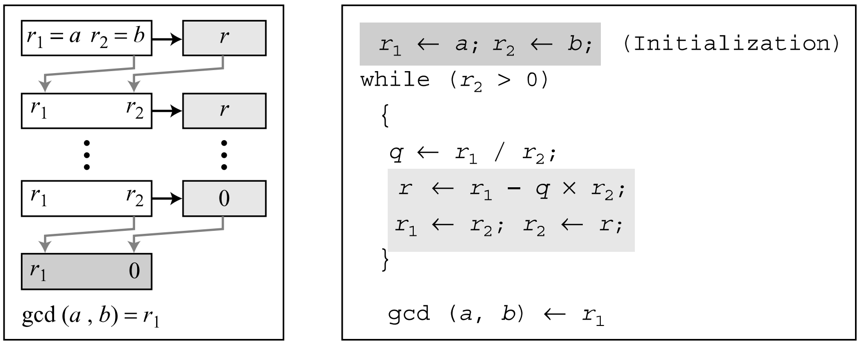
Fact 2: gcd $(a, b) =$ gcd (b, r), where r is the remainder of dividing a by b

The first fact tells us that if the second integer is 0, the greatest common divisor is the first one. The second fact allows us to change the value of a, b until b becomes 0. For example, to calculate the gcd (36, 10), we can use the second fact several times and the first fact once, as shown below.

$$\text{gcd}(36, 10) = \text{gcd}(10, 6) = \text{gcd}(6, 4) = \text{gcd}(4, 2) = \text{gcd}(2, 0) = 2$$

In other words, gcd (36, 10) = 2, gcd (10, 6) = 2, and so on. This means that instead of calculating gcd (36, 10), we can find gcd (2, 0). Figure 2.7 shows how we use the above two facts to calculate gcd (a, b).

Figure 2.7 *Euclidean algorithm*



```
$r_1 \leftarrow a;\ r_2 \leftarrow b;$   (Initialization)
while ($r_2 > 0$)
 {
  $q \leftarrow r_1 / r_2;$
   $r \leftarrow r_1 - q \times r_2;$
   $r_1 \leftarrow r_2;\ r_2 \leftarrow r;$
 }

  gcd ($a$, $b$) $\leftarrow r_1$
```

a. Process b. Algorithm

We use two variables, r_1 and r_2, to hold the changing values during the process of reduction. They are initialized to a and b. In each step, we calculate the remainder of r_1 divided by r_2 and store the result in the variable r. We then replace r_1 by r_2 and r_2 by r. The steps are continued until r_2 becomes 0. At this moment, we stop. The gcd (a, b) is r_1.

When gcd (a, b) = 1, we say that a and b are relatively prime.

Example 2.7

Find the greatest common divisor of 2740 and 1760.

Solution

We apply the above procedure using a table. We initialize r_1 to 2740 and r_2 to 1760. We have also shown the value of q in each step. We have gcd (2740, 1760) = 20.

q	r_1	r_2	r
1	2740	1760	980
1	1760	980	780
1	980	780	200
3	780	200	180
1	200	180	20
9	180	20	0
	20	0	

Example 2.8

Find the greatest common divisor of 25 and 60.

Solution

We chose this particular example to show that it does not matter if the first number is smaller than the second number. We immediately get our correct ordering. We have gcd (25, 65) = 5.

q	r_1	r_2	r
0	25	60	25
2	60	25	10
2	25	10	5
2	10	5	0
	5	0	

The Extended Euclidean Algorithm

Given two integers a and b, we often need to find other two integers, s and t, such that

$$s \times a + t \times b = \gcd(a, b)$$

The **extended Euclidean algorithm** can calculate the gcd (a, b) and at the same time calculate the value of s and t. The algorithm and the process is shown in Figure 2.8.

As shown in Figure 2.8, the extended Euclidean algorithm uses the same number of steps as the Euclidean algorithm. However, in each step, we use three sets of calculations and exchanges instead of one. The algorithm uses three sets of variables, r's, s's, and t's.