3.0 Modular Arithmetic

Modular arithmetic offers us a way to confine results to a particular range, just as the hours on a clock face confine us to reporting time relative to 12 or 24. We have seen in earlier chapters how, in some cryptographic applications, we want to perform some arithmetic operations on a plaintext character and guarantee that the result will be another character. Modular arithmetic enables us to do this; the results stay in the underlying range of numbers. An even more useful property is that the operations +, -, and * can be applied before or after the modulus is taken, with similar results.

Recall that a modulus applied to a nonnegative integer means remainder after division, so that 11 mod 3 = 2 since 11/3 = 3 with remainder 2. If a mod n = b then

a = c * n + b

for some integer c. Two different integers can have the same modulus: 11 mod 3 = 2 and 5 mod 3 = 2. Any two integers are equivalent under modulus n if their results mod n are equal. This property is denoted

x□ ny if and only if (x mod n) = (y mod n)

Equivalently,

x □ n y if and only if (x - y) = k * n for some k

In the following sections, unless we use parentheses to indicate otherwise, a modulus applies to a complete expression. Thus, you should interpret a + b mod n as (a + b) mod n, not a + (b mod n).

3.1 . Properties of Modular Arithmetic

Modular arithmetic on the nonnegative integers forms a construct called a **commutative ring** with operations + and * (addition and multiplication). Furthermore, if every number other than 0 has an inverse under *, the group is called a **Galois field.** All rings have the properties of associativity and distributivity; commutative rings, as their name implies, also have commutativity. Inverses under multiplication produce a Galois field. In particular, the integers mod a prime n are a Galois field. The properties of this arithmetic system are listed here.

Property	**Example**
associativity	a + (b + c) mod n = (a + b) + c mod n a * (b * c) mod n = (a * b)* c mod n

commutativity	$a + b \bmod n = b + a \bmod n$ $a * b \bmod n = b * a \bmod n$
distributivity	$a * (b + c) \bmod n = ((a * b) + (a * c)) \bmod n$
existence of identities	$a + 0 \bmod n = 0 + a \bmod n = a$ $a * 1 \bmod n = 1 * a \bmod n = a$
existence of inverses	$a + (-a) \bmod n = 0$ $a * (a-1) \bmod n = 1$ if a 0
reducibility	$(a + b) \bmod n = ((a \bmod n) + (b \bmod n)) \bmod n$ $(a * b) \bmod n = ((a \bmod n) * (b \bmod n)) \bmod n$

1.1 Example

As an example, consider the field of integers mod 5 shown in the tables below. These tables illustrate how to compute the sum or product of any two integers mod 5. However, the reducibility rule gives a method that you may find easier to use. To compute the sum or product of two integers mod 5, we compute the regular sum or product and then reduce this result by subtracting 5 until the result is between 0 and 4. Alternatively, we divide by 5 and keep only the remainder after division.

+	0	1	2	3	4
0	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	1
3	3	4	0	1	2
4	4	0	1	2	3

*	0	1	2	3	4
0	0	0	0	0	0

1	0	1	2	3	4
2	0	2	4	1	3
3	0	3	1	4	2
4	0	4	3	2	1

For example, let us compute 3 + 4 mod 5. Since 3 + 4 = 7 and 7 - 5 = 2, we can conclude that 3 + 4 mod 5 = 2. This fact is confirmed by the table. Similarly, to compute 4 * 4 mod 5, we compute 4 * 4 = 16. We can compute 16 - 5 = 11 - 5 = 6 - 5 = 1, or we can compute 16/5 = 3 with remainder 1. Either of these two approaches shows that 4 * 4 mod 5 = 1, as noted in the table. Since constructing the tables shown is difficult for large values of the modulus, the remainder technique is especially helpful.

3.2 Computing Inverses

In the ordinary system of multiplication on rational numbers, the inverse of any nonzero number a is 1/a, since a * (1/a) = 1. Finding inverses is not quite so easy in the finite fields just described. In this section we learn how to determine the multiplicative inverse of any element.

The inverse of any element a is that element b such that a * b = 1. The multiplicative inverse of a can be written a-1. Looking at the table for multiplication mod 5, we find that the inverse of 1 is 1, the inverse of 2 is 3 and, since multiplication is commutative, the inverse of 3 is also 2; finally, the inverse of 4 is 4. These values came from inspection, not from any systematic algorithm.

To perform one of the secure encryptions, we need a procedure for finding the inverse mod n of any element, even for very large values of n. An algorithm to determine a-1 directly is likely to be faster than a table search, especially for large values of n. Also, although there is a pattern to the elements in the table, it is not easy to generate the elements of a particular row, looking for a 1 each time we need an inverse. Fortunately, we have an algorithm that is reasonably simple to compute.

3.3 Fermat's Theorem

In number theory, Fermat's theorem states that for any prime p and any element a < p,

ap mod p = a

or

ap-1 mod p = 1
This result leads to the inverses we want. For a prime p and an element a < p, the inverse of a is that element x such that
ax mod p = 1
Combining the last two equations, we obtain
ax mod p = 1 = ap-1 mod p
so that
x = ap-2 mod p
This method is not a complete method for computing inverses, in that it works only for a prime p and an element a < p.

1.3.1 Example

We can use this formula to determine the inverse of 3 mod 5:

$$3-1 \bmod 5 = 35-2 \bmod 5$$

$$= 33 \bmod 5$$

$$= 27 \bmod 5$$

$$= 2$$

as we determined earlier from the multiplication table.

1.3.2 Algorithm for Computing Inverses

Another method to compute inverses is shown in the following algorithm. It is a fast approach that uses Euclid's algorithm for finding the greatest common divisor.
{**Compute x = a-1 mod n given a and n **}

```
c0:= n
c1:= a
b0:= 0
```

```
b1:= 1
i:= 1
repeat
    ci+1:= ci-1 mod ci
    t:= ci-1 div ci
    bi + 1:= bi-1 - t * bi
    i:= i + 1
until ci = 0
if (bi-1 0) then x:= bi-1 else x:= n + bi-1
```

3.4 The Chinese Remainder Theorem

One of the most useful results of number theory is the Chinese remainder theorem (CRT). In essence, the CRT says it is possible to reconstruct integers in a certain range from their residues modulo a set of pairwise relatively prime moduli.
The CRT is so called because it is believed to have been discovered by the Chinese mathematician Sun-Tsu in around 100 A.D.

Example:

The 10 integers in Z10, that is the integers 0 through 9, can be reconstructed from their two residues modulo 2 and 5 (the relatively prime factors of 10). Say the known residues of a decimal digit x are r2 = 0 and r5 = 3; that is, x mod 2 =0 and x mod 5 = 3. Therefore, x is an even integer in Z10 whose remainder, on division by 5, is 3. The unique solution is x = 8.

3.0 Modular Arithmetic

Modular arithmetic offers us a way to confine results to a particular range, just as the hours on a clock face confine us to reporting time relative to 12 or 24. We have seen in earlier chapters how, in some cryptographic applications, we want to perform some arithmetic operations on a plaintext character and guarantee that the result will be another character. Modular arithmetic enables us to do this; the results stay in the underlying range of numbers. An even more useful property is that the operations +, -, and * can be applied before or after the modulus is taken, with similar results.

Recall that a modulus applied to a nonnegative integer means remainder after division, so that 11 mod 3 = 2 since 11/3 = 3 with remainder 2. If a mod n = b then

a = c * n + b

for some integer c. Two different integers can have the same modulus: 11 mod 3 = 2 and 5 mod 3 = 2. Any two integers are equivalent under modulus n if their results mod n are equal. This property is denoted

x □ n y if and only if (x mod n) = (y mod n)

Equivalently,

x □ n y if and only if (x - y) = k * n for some k

In the following sections, unless we use parentheses to indicate otherwise, a modulus applies to a complete expression. Thus, you should interpret a + b mod n as (a + b) mod n, not a + (b mod n).

3.1. Properties of Modular Arithmetic

Modular arithmetic on the nonnegative integers forms a construct called a **commutative ring** with operations + and * (addition and multiplication). Furthermore, if every number other than 0 has an inverse under *, the group is called a **Galois field.** All rings have the properties of associativity and distributivity; commutative rings, as their name implies, also have commutativity. Inverses under multiplication produce a Galois field. In particular, the integers mod a prime n are a Galois field. The properties of this arithmetic system are listed here.

Property	**Example**
associativity	a + (b + c) mod n = (a + b) + c mod n a * (b * c) mod n = (a * b)* c mod n
commutativity	a + b mod n = b + a mod n a * b mod n = b * a mod n

distributivity $a * (b + c) \bmod n = ((a * b) + (a * c)) \bmod n$

existence of identities $a + 0 \bmod n = 0 + a \bmod n = a$

$a * 1 \bmod n = 1 * a \bmod n = a$

existence of inverses $a + (-a) \bmod n = 0$

$a * (a-1) \bmod n = 1$ if a 0

reducibility $(a + b) \bmod n = ((a \bmod n) + (b \bmod n)) \bmod n$

$(a * b) \bmod n = ((a \bmod n) * (b \bmod n)) \bmod n$

1.1 Example

As an example, consider the field of integers mod 5 shown in the tables below. These tables illustrate how to compute the sum or product of any two integers mod 5. However, the reducibility rule gives a method that you may find easier to use. To compute the sum or product of two integers mod 5, we compute the regular sum or product and then reduce this result by subtracting 5 until the result is between 0 and 4. Alternatively, we divide by 5 and keep only the remainder after division.

+	0	1	2	3	4
0	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	1
3	3	4	0	1	2
4	4	0	1	2	3

*	0	1	2	3	4
0	0	0	0	0	0

1	0	1	2	3	4
2	0	2	4	1	3
3	0	3	1	4	2
4	0	4	3	2	1

For example, let us compute 3 + 4 mod 5. Since 3 + 4 = 7 and 7 - 5 = 2, we can conclude that 3 + 4 mod 5 = 2. This fact is confirmed by the table. Similarly, to compute 4 * 4 mod 5, we compute 4 * 4 = 16. We can compute 16 - 5 = 11 - 5 = 6 - 5 = 1, or we can compute 16/5 = 3 with remainder 1. Either of these two approaches shows that 4 * 4 mod 5 = 1, as noted in the table. Since constructing the tables shown is difficult for large values of the modulus, the remainder technique is especially helpful.

3.2 Computing Inverses

In the ordinary system of multiplication on rational numbers, the inverse of any nonzero number a is 1/a, since a * (1/a) = 1. Finding inverses is not quite so easy in the finite fields just described. In this section we learn how to determine the multiplicative inverse of any element.

The inverse of any element a is that element b such that a * b = 1. The multiplicative inverse of a can be written a-1. Looking at the table for multiplication mod 5, we find that the inverse of 1 is 1, the inverse of 2 is 3 and, since multiplication is commutative, the inverse of 3 is also 2; finally, the inverse of 4 is 4. These values came from inspection, not from any systematic algorithm.

To perform one of the secure encryptions, we need a procedure for finding the inverse mod n of any element, even for very large values of n. An algorithm to determine a-1 directly is likely to be faster than a table search, especially for large values of n. Also, although there is a pattern to the elements in the table, it is not easy to generate the elements of a particular row, looking for a 1 each time we need an inverse. Fortunately, we have an algorithm that is reasonably simple to compute.

3.3 Fermat's Theorem

In number theory, Fermat's theorem states that for any prime p and any element a < p,

ap mod p = a

or

ap-1 mod p = 1

This result leads to the inverses we want. For a prime p and an element a < p, the inverse of a is that element x such that

ax mod p = 1

Combining the last two equations, we obtain

ax mod p = 1 = ap-1 mod p

so that

x = ap-2 mod p

This method is not a complete method for computing inverses, in that it works only for a prime p and an element a < p.

1.3.1 Example

We can use this formula to determine the inverse of 3 mod 5:

=35-2 mod 5

3-1 mod 5

= 33 mod 5

= 27 mod 5

= 2

as we determined earlier from the multiplication table.

1.3.2 Algorithm for Computing Inverses

Another method to compute inverses is shown in the following algorithm. It is a fast approach that uses Euclid's algorithm for finding the greatest common divisor.

{**Compute x = a-1 mod n given a and n **}

```
c0:= n
c1:= a
```

```
b0:= 0
b1:= 1
i:= 1
repeat
        ci+1:= ci-1 mod ci
        t:= ci-1 div ci
        bi + 1:= bi-1 - t * bi
        i:= i + 1
    until ci = 0
if (bi-1  0) then x:= bi-1 else x:= n + bi-1
```

3.4 Euler's Totient Function

Before presenting Euler's theorem, we need to introduce an important quantity in number theory, referred to as Euler's totient function and written f(n), defined as the number of positive integers less than n and relatively prime to n. By convention, f(1) = 1.

Determine f(37) and f(35).

Because 37 is prime, all of the positive integers from 1 through 36 are relatively prime to 37. Thus f(37) = 36.

To determine f(35), we list all of the positive integers less than 35 that are relatively prime to it:

1, 2, 3, 4, 6, 8, 9, 11, 12, 13, 16, 17, 18,

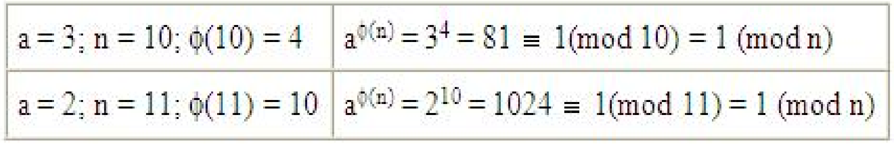
19, 22, 23, 24, 26, 27, 29, 31, 32, 33, 34.

There are 24 numbers on the list, so f(35) = 24.

1.4.1 Euler's Theorem

Euler's theorem states that for every *a* and *n* that are relatively prime:

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



a = 3; n = 10; ϕ(10) = 4	$a^{\phi(n)} = 3^4 = 81 \equiv 1(\bmod 10) = 1 (\bmod n)$
a = 2; n = 11; ϕ(11) = 10	$a^{\phi(n)} = 2^{10} = 1024 \equiv 1(\bmod 11) = 1 (\bmod n)$

3.5 The EuclideanAlgorithm

The *Euclidean Algorithm* is used to discover the greatest common divisor of two integers. In cryptography, it is most often used to determine if two integers are co-prime, i.e.

$$\gcd(a,b)=1$$

- .

In order to find $\gcd(a,b)=1$ where $a > b$, efficiently when working with very large numbers, as with cryptosystems, a method exists to do so. The Euclidean algorithm operates as follows - First, divide a by b, writing the quotient q_1, and the remainder r_1. Note this can be written in equation form as $a = q_1 b + r_1$. Next perform the same operation using $b$in □'s place: $b = q_2 r_1 + r_2$. Continue with this pattern until the final remainder is zero. Numerical examples and a formal algorithm follow which should make this inherent pattern clear.

3.5.1 Mathematical Description

$$
\begin{aligned}
a &= q_1 b + r_1 \\
b &= q_2 r_1 + r_2 \\
r_1 &= q_3 r_2 + r_3 \\
r_2 &= q_4 r_3 + r_4 \\
&\vdots \\
r_{n-2} &= q_n r_{n-1} + r_n
\end{aligned}
$$

When $r_n = 0$, stop with $\gcd(a, b) = r_{n-1}$.

3.5.2 Numerical Examples

Example 1 - To find gcd(17,043,12,660)

17,043 = 1 · 12,660 + 4383

12,660 = 2 · 4,383 + 3894

4,383 = 1 · 3,894 + 489

3,894 = 7 · 489 + 471

489 = 1 · 471 + 18

471 = 26 · 18 + 3

18 = 6 · 3 + 0

gcd (17,043,12,660) = 3

Example 2 - To find gcd(2,008,1,963)

2,008 = 1 · 1,963 + 45

1,963 = 43 · 45 + 28

45 = 1 · 28 + 17

28 = 1 · 17 + 11

17 = 1 · 11 + 6

11 = 1 · 6 + 5

6 = 1 ▫ 5 + 1

5 = 5 ▫ 1 + 0

gcd (2,008,1963) = 1 Note: the two number are co-prime.

3.6 The Chinese Remainder Theorem

If one wants to solve a system of congruences with different moduli, it is possible to do so as follows:

$x \equiv a_1 \bmod m_1$
$x \equiv a_2 \bmod m_2$

. . .

$x \equiv a_k \bmod m_k$

A simultaneous solution □exists if and only if $\gcd(m_i, m_j) = 1$ with $(i \neq j)$, and any two solutions are congruent to one another modulo $M = m_1 m_2 \ldots m_k$.

The steps for finding the simultaneous solution using the Chinese Remainder theorem are as follows:

1. Compute □
2. Compute $M_i = M/m_i$ for each of the different □'s
3. Find the inverse N of $M_i \bmod m_i$ for each □ using the Extended Euclidean algorithm
4. Multiply out $a_i M_i N_i$for each □
5. Sum all $a_i M_i N_i$
6. Compute $\sum_{i=1}^{k} a_i M_i N_i \bmod M$ to obtain the least nonnegative residue

Example

Given:

$$\begin{aligned}x &\equiv 1 \bmod 11\\ x &\equiv 2 \bmod 7\\ x &\equiv 3 \bmod 5\\ x &\equiv 4 \bmod 9\end{aligned}$$

$$M = 3465$$

$$\begin{aligned}M_{11} &= 315\\ M_7 &= 495\\ M_5 &= 693\\ M_9 &= 385\end{aligned}$$

Using the Extended Euclidean algorithm:

$$\begin{aligned}315N &\equiv 1 \bmod 11 \quad N = -3\\ 315N &\equiv 1 \bmod 7 \quad N = 3\\ 315N &\equiv 1 \bmod 5 \quad N = 2\\ 315N &\equiv 1 \bmod 9 \quad N = 4\end{aligned}$$

$$\sum_{i=1}^{4} = \begin{cases} 1 \cdot 315 \cdot (-3) = -945 \\ 2 \cdot 495 \cdot 3 = 2970 \\ 3 \cdot 639 \cdot 2 = 4158 \\ 4 \cdot 385 \cdot 4 = 6160 \end{cases}$$

$$\sum = 12343$$

$$x = 12343 \bmod 3465 = 1948$$

One of the most useful results of number theory is the Chinese remainder theorem (CRT). In essence, the CRT says it is possible to reconstruct integers in a certain range from their residues modulo a set of pairwise relatively prime moduli. The CRT is so called because it is believed to have been discovered by the Chinese mathematician Sun-Tsu in around 100 A.D.

The 10 integers in Z10, that is the integers 0 through 9, can be reconstructed from their two residues modulo 2 and 5 (the relatively prime factors of 10). Say the known residues of a decimal digit x are r2 = 0 and r5 = 3;

that is, x mod 2 =0 and x mod 5 = 3. Therefore, x is an even integer in Z10 whose remainder, on division by 5, is 3. The unique solution is x = 8.

Example:

In considering the problem of finding modular square roots, we found that the problem for a general modulus *m* could be reduced to that for a prime power modulus. The next problem would be how to piece the solutions for prime powers together to solve the original congruence. This is done by the Chinese Remainder Theorem, so-called because it appeared in ancient Chinese manuscripts.

A typical problem is to find integers *x* that simultaneously solve

$$x \equiv 13 \pmod{27}$$

$$x \equiv 7 \pmod{16}$$

It's important in our applications that the two moduli be relatively prime; otherwise, we would have to check that the two congruences are consistent. The Chinese Remainder Theorem has a very simple answer:

Chinese Remainder Theorem: For relatively prime moduli *m* and *n*, the congruences

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

$$x \equiv b \pmod{n}$$

have a unique solution *x* modulo *mn*.

Our example problem would have a unique solution modulo $16 \cdot 27 = 432$.

It's better than this; there is a relatively simple algorithm to find the solution. Since all number theory algorithms ultimately come down to Euclid's algorithm, you can be sure it happens here as well.

First let's consider an even simpler example. Suppose we want all numbers *x* that satisfy

$$x \equiv 2 \pmod{3}$$

$$x \equiv 3 \pmod{5}$$

The numbers that satisfy the first congruence are in the sequence

$$2,\ 5,\ 8,\ 11,\ 14,\ 17, \ldots$$

Just scan this sequence for a term that also leaves remainder 3 after division by 5. The answer is *x*=8.

Euclid's algorithm can be used to solve several problems: finding the greatest common divisor *d* of two numbers *m* and *n*, and finding two numbers *x* and *y* such that *mx*+*ny*=*d*, and solving congruences $ax \equiv b \pmod{m}$ for *x*. We will now see how it also helps to solve Chinese Remainder problems. We will start with *m*=27 and *n*=16. Here is what Euclid's algorithm gives.

							0	1
Dividend	=	Quotient	·	Divisor	+	Remainder	1	0
27	=	1	·	16	+	11	1	1
16	=	1	·	11	+	5	2	1
11	=	2	·	5	+	1	5	3
5	=	5	·	1	+	0	27	16

From the theory , we know that $5 \cdot 16 - 3 \cdot 27 = -1$, or equivalently that $3 \cdot 27 - 5 \cdot 16 = 1$. This equation gives a very easy solution to our original simultaneous congruences:

$$x = (3 \cdot 27)\,7 - (5 \cdot 16)\,13 = -473.$$

Without working with -473, it is possible to quickly check that $-473 \equiv 13 \pmod{27}$ and $-473 \equiv 7 \pmod{16}$. The reason lies in the equation $3 \cdot 27 - 5 \cdot 16 = 1$. If we first interpret that equation mod 27, we see that $-5 \cdot 16 \equiv 1 \pmod{27}$. Multiplying that congruence by 13 gives $-5 \cdot 16 \cdot 13 \equiv 13 \pmod{27}$. Similarly, if we interpret things modulo 16, we find that $3 \cdot 27 \cdot 7 \equiv 7 \pmod{16}$. That verifies our solution for the Chinese Reminder problem.

The number *x*=-473 is not the only solution; if we modify it by a multiple of both 27 and 16, we won't change the congruences. That means $-473 + k\,432$ is a solution for any integer *k*. Often we add a multiple of 432 to make the solution positive: $-473 + 2 \cdot 432 = 391$. We can check that $391 \equiv 13 \pmod{27}$ and $391 \equiv 7 \pmod{16}$.

Here is a summary of the whole process for

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

$$x \equiv b \pmod{n}$$

First find integers *u* and *v* such that

mu+*nv*=1

Then all solutions are

$$x \equiv (mu)b + (nv)a \pmod{mn}$$

One more example: $x \equiv 23 \pmod{100}$ and $x \equiv 31 \pmod{49}$. First we have to solve

100*u*+49 *v*=1

Euclid's algorithm gives

								0	1
Dividend	=	Quotient	·	Divisor	+	Remainder		1	0
100	=	2	·	49	+	2		2	1
49	=	24	·	2	+	1		49	24
2	=	2	·	1	+	0		100	49

Then, $49 \cdot 49 - 24 \cdot 100 = 1$. The solution is

$$49 \cdot 49 \cdot 23 - 24 \cdot 100 \cdot 31 = -19177 \equiv 423 \pmod{4900}$$

Back to Top