CHAPTER 4

Mathematics of Cryptography

Part II: Algebraic Structures

Objectives

This chapter prepares the reader for the next few chapters, which will discuss modern symmetric-key ciphers based on algebraic structures. This chapter has several objectives:

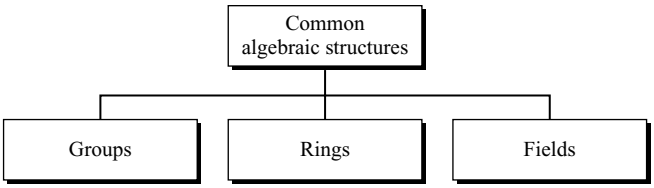
- ❏ To review the concept of algebraic structures
- ❏ To define and give some examples of groups
- ❏ To define and give some examples of rings
- ❏ To define and give some examples of fields
- ❏ To emphasize the finite fields of type GF(2^n) that make it possible to perform operations such as addition, subtraction, multiplication, and division on *n*-bit words in modern block ciphers

The next few chapters will discuss modern symmetric-key block ciphers that perform some operations on *n*-bit words. Understanding and analyzing these ciphers requires some knowledge of a branch of modern algebra called algebraic structures. This chapter first reviews the topic of algebraic structures, and then it shows how to perform operations such as addition or multiplication on *n*-bit words.

4.1 ALGEBRAIC STRUCTURES

Chapter 2 discussed some sets of numbers, such as $\mathbf{Z}$, $\mathbf{Z}_n$, $\mathbf{Z}_n^*$, $\mathbf{Z}_p$ and $\mathbf{Z}_p^*$. Cryptography requires sets of integers and specific operations that are defined for those sets. The combination of the set and the operations that are applied to the elements of the set is called an **algebraic structure.** In this chapter, we will define three common algebraic structures: *groups, rings,* and *fields* (Figure 4.1).

Figure 4.1 *Common algebraic structures*



Groups

A **group** (**G**) is a set of elements with a binary operation "•" that satisfies four properties (or axioms). A **commutative group,** also called an **abelian group,** is a group in which the operator satisfies the four properties for groups plus an extra property, commutativity. The four properties for groups plus commutativity are defined as follows:

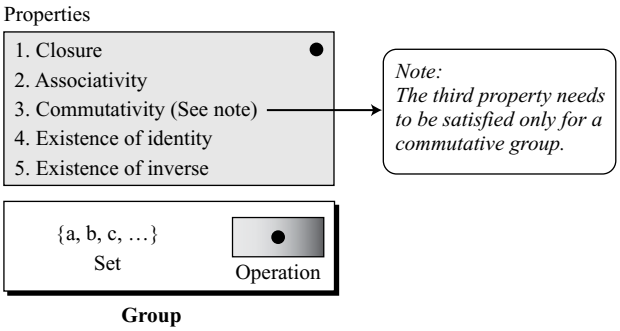
- **Closure:** If a and b are elements of **G**, then c = $a \bullet b$ is also an element of **G**. This means that the result of applying the operation on any two elements in the set is another element in the set.
- **Associativity:** If a, b, and c are elements of **G**, then $(a \bullet b) \bullet c = a \bullet (b \bullet c)$. In other words, it does not matter in which order we apply the operation on more than two elements.
- **Commutativity:** For all a and b in **G**, we have $a \bullet b = b \bullet a$. Note that this property needs to be satisfied only for a commutative group.
- **Existence of identity:** For all a in **G**, there exists an element e, called the identity element, such that $e \bullet a = a \bullet e = a$.
- **Existence of inverse:** For each a in **G**, there exists an element a', called the inverse of a, such that $a \bullet a' = a' \bullet a = e$.

Figure 4.2 shows the concept of a group.

Application

Although a group involves a single operation, the properties imposed on the operation allow the use of a pair of operations as long as they are inverses of each other. For example, if the defined operation is addition, the group supports both addition and subtraction, because subtraction is addition using the additive inverse. This is also true for multiplication and division. However, a group can support only addition/subtraction or multiplication/division operations, but not the both at the same time.

Figure 4.2 *Group*



Example 4.1

The set of residue integers with the addition operator, $\mathbf{G} = \langle \mathbf{Z}_n, + \rangle$, is a commutative group. We can perform addition and subtraction on the elements of this set without moving out of the set. Let us check the properties.

1. Closure is satisfied. The result of adding two integers in $\mathbf{Z}_n$ is another integer in $\mathbf{Z}_n$.
2. Associativity is satisfied. The result of $4 + (3 + 2)$ is the same as $(4 + 3) + 2$.
3. Commutativity is satisfied. We have $3 + 5 = 5 + 3$.
4. The identify element is 0. We have $3 + 0 = 0 + 3 = 3$.
5. Every element has an additive inverse. The inverse of an element is its complement. For example, the inverse of 3 is -3 ($n - 3$ in $\mathbf{Z}_n$) and the inverse of -3 is 3. The inverse allows us to perform subtraction on the set.

Example 4.2

The set $\mathbf{Z}_n^*$ with the multiplication operator, $\mathbf{G} = \langle \mathbf{Z}_n^*, \times \rangle$, is also an abelian group. We can perform multiplication and division on the elements of this set without moving out of the set. It is easy to check the first three properties. The identity element is 1. Each element has an inverse that can be found according to the extended Euclidean algorithm.

Example 4.3

Although we normally think about a group as the set of numbers with the regular operations such as addition or subtraction, the definition of the group allows us to define any set of objects and an operation that satisfies the above-mentioned properties. Let us define a set $\mathbf{G} = \langle \{a, b, c, d\}, \bullet \rangle$ and the operation as shown in Table 4.1.

Table 4.1 *Operation table for Example 4.3*

•	*a*	*b*	*c*	*d*
a	*a*	*b*	*c*	*d*
b	*b*	*c*	*d*	*a*
c	*c*	*d*	*a*	*b*
d	*d*	*a*	*b*	*c*

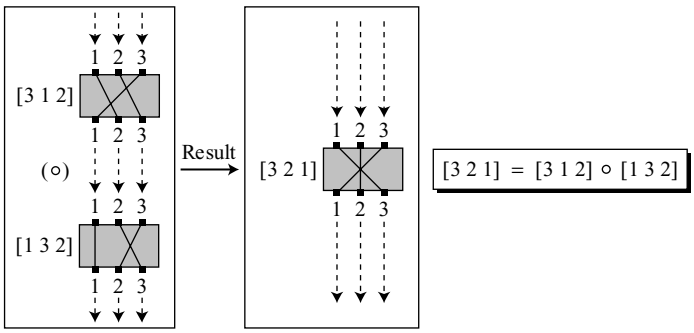
This is an abelian group. All five properties are satisfied:

1. Closure is satisfied. Applying the operation on any pair of elements result in another elements in the set.
2. Associativity is also satisfied. To prove this we need to check the property for any combination of three elements. For example, $(a + b) + c = a + (b + c) = d$.
3. The operation is commutative. We have $a + b = b + a$.
4. The group has an identity element, which is a.
5. Each element has an inverse. The inverse pairs can be found by finding the identity in each row (shaded). The pairs are (a, a), (b, d), (c, c).

Example 4.4

In a group, the elements in the set do not have to be numbers or objects; they can be rules, mappings, functions, or even actions. A very interesting group is the **permutation group.** The set is the set of all permutations, and the operation is composition: applying one permutation after another. Figure 4.3 shows composition of two permutations that transpose three inputs to create three outputs.

Figure 4.3 *Composition of permutations (Example 4.4)*



The inputs and outputs can be characters (Chapter 2) or can be bits (Chapter 5). We have shown each permutation by a table in which the content shows where the input comes from and the index (not shown) defines the output. Composition involve applying two permutations, one after the other. Note that the expression in Figure 4.3 is read from right to left: the first permutation is [1 3 2] followed by [3 1 2]; the result is [3 2 1]. With three inputs and three outputs, there can be 3! or 6 different permutations. Table 4.2 shows how the operation is defined. The first row is the first permutation; the first column is the second permutation. The result is the cross-section element.

In this case, only four properties are satisfied; the group is non-abelian.

1. Closure is satisfied.
2. Associativity is also satisfied. To prove this we need to check the property for any combination of three elements.
3. The commutative property is not satisfied. This can be easily checked, but we leave it as an exercise.

Table 4.2 *Operation table for permutation group*

°	[1 2 3]	[1 3 2]	[2 1 3]	[2 3 1]	[3 1 2]	[3 2 1]
[1 2 3]	[1 2 3]	[1 3 2]	[2 1 3]	[2 3 1]	[3 1 2]	[3 2 1]
[1 3 2]	[1 3 2]	[1 2 3]	[2 3 1]	[2 1 3]	[3 2 1]	[3 1 2]
[2 1 3]	[2 1 3]	[3 1 2]	[1 2 3]	[3 2 1]	[1 3 2]	[2 3 1]
[2 3 1]	[2 3 1]	[3 2 1]	[1 3 2]	[3 1 2]	[1 2 3]	[2 1 3]
[3 1 2]	[3 1 2]	[2 1 3]	[3 2 1]	[1 2 3]	[2 3 1]	[1 3 2]
[3 2 1]	[3 2 1]	[2 3 1]	[3 1 2]	[1 3 2]	[2 1 3]	[1 2 3]

4. The set has an identity element, which is [1 2 3] (no permutation). These are shaded.
5. Each element has an inverse. The inverse pairs can be found using the identity elements.

Example 4.5

In the previous example, we showed that a set of permutations with the composition operation is a group. This implies that using two permutations one after another cannot strengthen the security of a cipher, because we can always find a permutation that can do the same job because of the closure property.

Finite Group

A group is called a **finite group** if the set has a finite number of elements; otherwise, it is an **infinite group.**

Order of a Group

The **order of a group,** $|\mathbf{G}|$, is the number of elements in the group. If the group is not finite, its order is infinite; if the group is finite, the order is finite.

Subgroups

A subset **H** of a group **G** is a **subgroup** of **G** if **H** itself is a group with respect to the operation on **G.** In other words, if $\mathbf{G} = \langle \mathbf{S}, \bullet \rangle$ is a group, $\mathbf{H} = \langle \mathbf{T}, \bullet \rangle$ is a group under the same operation, and **T** is a nonempty subset of **S,** then **H** is a subgroup of **G.** The above definition implies that:

1. If a and b are members of both groups, then $c = a \bullet b$ is also a member of both groups.
2. The group share the same identity element.
3. If a is a member of both groups, the inverse of a is also a member of both groups.
4. The group made of the identity element of **G, H** $= \langle \{e\}, \bullet \rangle$, is a subgroup of **G.**
5. Each group is a subgroup of itself.

Example 4.6

Is the group $\mathbf{H} = \langle \mathbf{Z}_{10}, + \rangle$ a subgroup of the group $\mathbf{G} = \langle \mathbf{Z}_{12}, + \rangle$?

Solution

The answer is no. Although **H** is a subset of **G,** the operations defined for these two groups are different. The operation in **H** is addition modulo 10; the operation in **G** is addition modulo 12.

Cyclic Subgroups

If a subgroup of a group can be generated using the power of an element, the subgroup is called the **cyclic subgroup.** The term *power* here means repeatedly applying the group operation to the element:

$$a^n \rightarrow a \bullet a \bullet \ldots \bullet a \quad (n \text{ times})$$

The set made from this process is referred to as <*a*>. Note that the duplicate elements must be discarded. Note also that $a^0 = e$.

Example 4.7

Four cyclic subgroups can be made from the group $\mathbf{G} = \langle \mathbf{Z}_6, + \rangle$. They are $\mathbf{H_1} = \langle \{0\}, + \rangle$, $\mathbf{H_2} = \langle \{0, 2, 4\}, + \rangle$, $\mathbf{H_3} = \langle \{0, 3\}, + \rangle$, and $\mathbf{H_4} = \mathbf{G}$. Note that when the operation is addition, a^n means multiplying n by a. Note also that in all of these groups, the operation is addition modulo 6. The following show how we find the elements of these cyclic subgroups.

a. The cyclic subgroup generated from 0 is $\mathbf{H}_1$, which has only one element, the identity element.

$0^0 \bmod 6 = 0$ (stop: the process will be repeated)

b. The cyclic subgroup generated from 1 is $\mathbf{H}_4$, which is **G** itself.

$1^0 \bmod 6 = 0$
$1^1 \bmod 6 = 1$
$1^2 \bmod 6 = (1 + 1) \bmod 6 = 2$
$1^3 \bmod 6 = (1 + 1 + 1) \bmod 6 = 3$
$1^4 \bmod 6 = (1 + 1 + 1 + 1) \bmod 6 = 4$
$1^5 \bmod 6 = (1 + 1 + 1 + 1 + 1) \bmod 6 = 5$ (stop: the process will be repeated)

c. The cyclic subgroup generated from 2 is $\mathbf{H}_2$, which has three elements: 0, 2, and 4.

$2^0 \bmod 6 = 0$
$2^1 \bmod 6 = 2$
$2^2 \bmod 6 = (2 + 2) \bmod 6 = 4$ (stop: the process will be repeated)

d. The cyclic subgroup generated from 3 is $\mathbf{H}_3$, which has two elements: 0 and 3.

$3^0 \bmod 6 = 0$
$3^1 \bmod 6 = 3$ (stop: the process will be repeated)

e. The cyclic subgroup generated from 4 is $\mathbf{H}_2$; this is not a new subgroup.

$4^0 \bmod 6 = 0$
$4^1 \bmod 6 = 4$
$4^2 \bmod 6 = (4 + 4) \bmod 6 = 2$ (stop: the process will be repeated)

f. The cyclic subgroup generated from 5 is $\mathbf{H}_4$, which is **G** itself.

$5^0 \bmod 6 = 0$
$5^1 \bmod 6 = 5$
$5^2 \bmod 6 = 4$
$5^3 \bmod 6 = 3$
$5^4 \bmod 6 = 2$
$5^5 \bmod 6 = 1$ (stop: the process will be repeated)

Example 4.8

Three cyclic subgroups can be made from the group $\mathbf{G} = \langle \mathbf{Z}_{10}^*, \times \rangle$. **G** has only four elements: 1, 3, 7, and 9. The cyclic subgroups are $\mathbf{H_1} = \langle \{1\}, \times \rangle$, $\mathbf{H_2} = \langle \{1, 9\}, \times \rangle$, and $\mathbf{H_3} = \mathbf{G}$. The following show how we find the elements of these subgroups.

a. The cyclic subgroup generated from 1 is $\mathbf{H}_1$. The subgroup has only one element, the identity element.

$1^0 \bmod 10 = 1$ (stop: the process will be repeated)

b. The cyclic subgroup generated from 3 is $\mathbf{H}_3$, which is **G** itself.

$3^0 \bmod 10 = 1$
$3^1 \bmod 10 = 3$
$3^2 \bmod 10 = 9$
$3^3 \bmod 10 = 7$ (stop: the process will be repeated)

c. The cyclic subgroup generated from 7 is $\mathbf{H}_3$, which is **G** itself.

$7^0 \bmod 10 = 1$
$7^1 \bmod 10 = 7$
$7^2 \bmod 10 = 9$
$7^3 \bmod 10 = 3$ (stop: the process will be repeated)

d. The cyclic subgroup generated from 9 is $\mathbf{H}_2$. The subgroup has only two elements.

$9^0 \bmod 10 = 1$
$9^1 \bmod 10 = 9$ (stop: the process will be repeated)

Cyclic Groups

A **cyclic group** is a group that is its own cyclic subgroup. In Example 4.7, the group **G** has a cyclic subgroup $\mathbf{H_5} = \mathbf{G}$. This means that the group **G** is a cyclic group. In this case, the element that generates the cyclic subgroup can also generate the group itself. This element is referred to as a generator. If g is a generator, the elements in a finite cyclic group can be written as

$$\{e, g, g^2, \ldots, g^{n-1}\}, \text{ where } g^n = e$$

Note that a cyclic group can have many generators.

Example 4.9

a. The group $\mathbf{G} = <\mathbf{Z}_6, +>$ is a cyclic group with two generators, $g = 1$ and $g = 5$.
b. The group $\mathbf{G} = <\mathbf{Z}_{10}^*, \times>$ is a cyclic group with two generators, $g = 3$ and $g = 7$.

Lagrange's Theorem

Lagrange's theorem relates the order of a group to the order of its subgroup. Assume that **G** is a group, and **H** is a subgroup of **G.** If the order of **G** and **H** are $|\mathbf{G}|$ and $|\mathbf{H}|$, respectively, then, based on this theorem, $|\mathbf{H}|$ divides $|\mathbf{G}|$. In Example 4.7, $|\mathbf{G}| = 6$. The order of the subgroups are $|\mathbf{H}_1| = 1$, $|\mathbf{H}_2| = 3$, $|\mathbf{H}_3| = 2$, and $|\mathbf{H}_4| = 6$. Obviously all of these orders divide 6.

Lagrange's theorem has a very interesting application. Given a group **G** of order $|\mathbf{G}|$, the orders of the potential subgroups can be easily determined if the divisors of $|\mathbf{G}|$ can be found. For example, the order of the group $\mathbf{G} = <\mathbf{Z}_{17}, +>$ is 17. The only divisors of 17 are 1 and 17. This means that this group can have only two subgroups, $\mathbf{H}_1$ with the identity element and $\mathbf{H}_2 = \mathbf{G}$.

Order of an Element

The **order of an element** a in a group, ord(a), is the smallest integer n such that $a^n = e$. The definition can be paraphrased: the order of an element is the order of the cyclic group it generates.

Example 4.10

a. In the group $\mathbf{G} = <\mathbf{Z}_6, +>$, the orders of the elements are: ord(0) = 1, ord(1) = 6, ord(2) = 3, ord(3) = 2, ord(4) = 3, ord(5) = 6.
b. In the group $\mathbf{G} = <\mathbf{Z}_{10}^*, \times>$, the orders of the elements are: ord(1) = 1, ord(3) = 4, ord(7) = 4, ord(9) = 2.

Ring

A **ring,** denoted as $\mathbf{R} = <\{\ldots\}, \bullet, \square>$, is an algebraic structure with two operations. The first operation must satisfy all five properties required for an abelian group. The second operation must satisfy only the first two. In addition, the second operation must be distributed over the first. **Distributivity** means that for all a, b, and c elements of **R,** we have $a \square (b \bullet c) = (a \square b) \bullet (a \square c)$ and $(a \bullet b) \square c = (a \square c) \bullet (b \square c)$. A **commutative ring** is a ring in which the commutative property is also satisfied for the second the operation. Figure 4.4 shows a ring and a commutative ring.

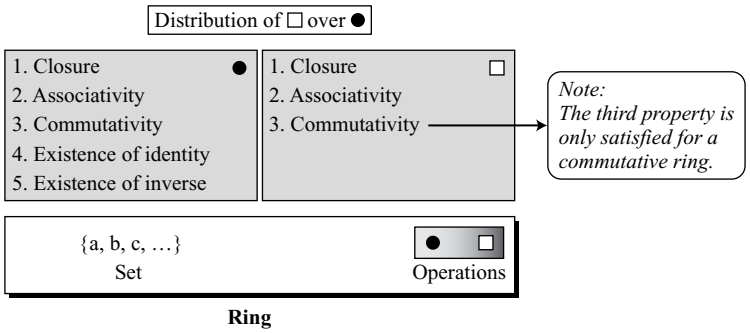
Application

A ring involves two operations. However, the second operation can fail to satisfy the third and fourth properties. In other words, the first operation is actually a pair of operation such as addition and subtraction; the second operation is a single operation, such as multiplication, but not division.

Example 4.11

The set **Z** with two operations, addition and multiplication, is a commutative ring. We show it by $\mathbf{R} = <\mathbf{Z}, +, \times>$. Addition satisfies all of the five properties; multiplication satisfies only three

Figure 4.4 *Ring*

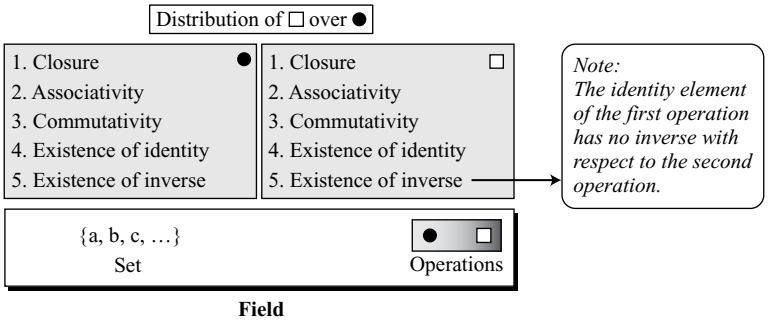


properties. Multiplication also distributes over addition. For example, $5 \times (3 + 2) = (5 \times 3) + (5 \times 2) = 25$. Although, we can perform addition and subtraction on this set, we can perform only multiplication, but not division. Division is not allowed in this structure because it yields an element out of the set. The result of dividing 12 by 5 is 2.4, which is not in the set.

Field

A **field,** denoted by $\mathbf{F} = <\{\ldots\}, \bullet, \square>$ is a commutative ring in which the second operation satisfies all five properties defined for the first operation except that the identity of the first operation (sometimes called the zero element) has no inverse. Figure 4.5 shows the field.

Figure 4.5 *Field*



Application

A field is a structure that supports two pairs of operations that we have used in mathematics: addition/subtraction and multiplication/division. There is one exception: division by zero is not allowed.

Finite Fields

Although we have fields of infinite order, only finite fields extensively used in cryptography. A **finite field,** a field with a finite number of elements, are very important structures in cryptography. Galois showed that for a field to be finite, the number of elements should be p^n, where p is a prime and n is a positive integer. The finite fields are usually called **Galois fields** and denoted as $\mathbf{GF}(p^n)$.

A Galois field, $\mathbf{GF}(p^n)$, is a finite field with p^n elements.

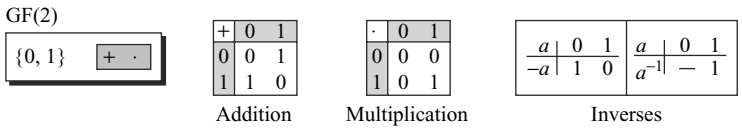
GF(p) Fields

When $n = 1$, we have $\mathbf{GF}(p)$ field. This field can be the set $\mathbf{Z}_p$, $\{0, 1, \ldots, p - 1\}$, with two arithmetic operations (addition and multiplication). Recall that in this set each element has an additive inverse and that nonzero elements have a multiplicative inverse (no multiplicative inverse for 0).

Example 4.12

A very common field in this category is **GF**(2) with the set {0, 1} and two operations, addition and multiplication, as shown in Figure 4.6.

Figure 4.6 *GF(2) field*



+	0	1
0	0	1
1	1	0

·	0	1
0	0	0
1	0	1

a	0	1
$-a$	1	0

a	0	1
a^{-1}	—	1

There are several things to notice about this field. First, the set has only two elements, which are binary digits or bits (0 and 1). Second, the addition operation is actually the exclusive-or (XOR) operation we use on two binary digits. Third, the multiplication operation is the AND operation we use on two binary digits. Fourth, addition and subtraction operations are the same (XOR operation). Fifth, multiplication and division operations are the same (AND operation).

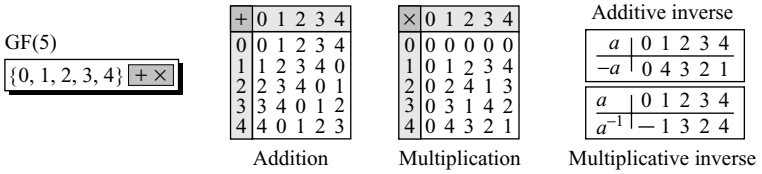
Addition/subtraction in GF(2) is the same as the XOR operation; multiplication/division is the same as the AND operation.

Example 4.13

We can define **GF**(5) on the set $\mathbf{Z}_5$ (5 is a prime) with addition and multiplication operators as shown in Figure 4.7.

Although we can use the extended Euclidean algorithm to find the multiplicative inverses of elements in **GF**(5), it is simpler to look at the multiplication table and find each pair with the product equal to 1. They are (1,1), (2, 3), (3, 2), and (4, 4). Note that we can apply addition/subtraction and multiplication/division on the set except that division by 0 is not allowed.

Figure 4.7 *GF(5) field*



GF(p^n) Fields

In addition to GF(p) fields, we are also interested in GF(p^n) fields in cryptography. However, the set $\mathbf{Z}$, $\mathbf{Z}_n$, $\mathbf{Z}_n$* and $\mathbf{Z}_p$, which we have used so far with operations such as addition and multiplication, cannot satisfy the requirement of a field. Some new sets and some new operations on those sets must be defined. The next section, we shows how **GF**(2^n) is a very useful field in cryptography.

Summary

The study of three algebraic structures allows us to use sets in which operations similar to addition/subtraction and multiplication/division can be used with the set. We need to distinguish between the three structures. The first structure, the group, supports one related pair of operations. The second structure, the ring, supports one related pair of operations and one single operation. The third structure, the field, supports two pairs of operations. Table 4.3 may help us to see the difference.

Table 4.3 *Summary of algebraic structures*

Algebraic Structure	*Supported Typical Operations*	*Supported Typical Sets of Integers*
Group	(+ −) or (× ÷)	$\mathbf{Z}_n$ or $\mathbf{Z}_n$*
Ring	(+ −) and (×)	$\mathbf{Z}$
Field	(+ −) and (× ÷)	$\mathbf{Z}_p$

4.2 GF(2^n) FIELDS

In cryptography, we often need to use four operations (addition, subtraction, multiplication, and division). In other words, we need to use fields. However, when we work with computers, the positive integers are stored in the computer as n-bit words in which n is usually 8, 16, 32, 64, and so on. This means that the range of integers is 0 to 2^n-1. The modulus is 2^n. So we have two choices if we want to use a field:

1. We can use **GF**(p) with the set $\mathbf{Z}_p$, where p is the largest prime number less than 2^n. Although this scheme works, it is inefficient because we cannot use the integers from p to $2^n - 1$. For example, if $n = 4$, the largest prime less than 2^4 is 13. This means that we cannot use integers 13, 14, and 15. If $n = 8$, the largest prime less than 2^8 is 251, so we cannot use 251, 252, 253, 254, and 255.

2. We can work in GF(2^n) and uses a set of 2^n elements. The elements in this set are n-bit words. For example, if $n = 3$, the set is

$$\{000, 001, 010, 011, 100, 101, 110, 111\}$$

However, we cannot interpret each element as an integer between 0 to 7 because the regular four operations cannot be applied (the modulus 2^n is not a prime). We need to define a set of n-bit words and two new operations that satisfies the properties defined for a field.

Example 4.14

Let us define a GF(2^2) field in which the set has four 2-bit words: {00, 01, 10, 11}. We can redefine addition and multiplication for this field in such a way that all properties of these operations are satisfied, as shown in Figure 4.8.

Figure 4.8 *An example of a GF(2^2) field*

Addition

⊕	00	01	10	11
00	00	01	10	11
01	01	00	11	10
10	10	11	00	01
11	11	10	01	00

Identity: 00

Multiplication

⊗	00	01	10	11
00	00	00	00	00
01	00	01	10	11
10	00	10	11	01
11	00	11	01	10

Identity: 01

Each word is the additive inverse of itself. Every word (except 00) has a multiplicative inverse. The multiplicative inverse pairs are (01, 01) and (10, 11). Addition and multiplication are defined in terms of polynomials.

Polynomials

Although we can directly define the rules for addition and multiplication operations on n-bit words that satisfy the properties in GF(2^n), it is easier to work with a representation of n-bit words, a polynomial of degree $n - 1$. A **polynomial** of degree $n - 1$ is an expression of the form

$$f(x) = a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \cdots + a_1x^1 + a_0x^0$$

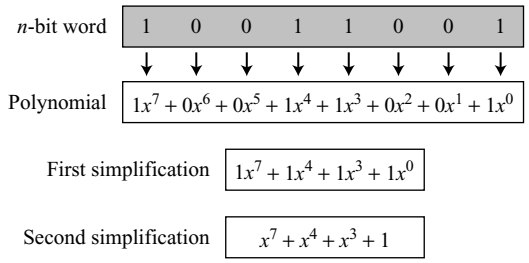
where x^i is called the ith term and a_i is called coefficient of the ith term. Although we are familiar with polynomials in algebra, to represent an n-bit word by a polynomial we need to follow some rules:

a. The power of x defines the position of the bit in the n-bit word. This means the leftmost bit is at position zero (related to x^0); the rightmost bit is at position $n - 1$ (related to x^{n-1}).
b. The coefficients of the terms define the value of the bits. Because a bit can have only a value of 0 or 1, our polynomial coefficients can be either 0 or 1.

Example 4.15

Figure 4.9 show how we can represent the 8-bit word (10011001) using a polynomials.

Figure 4.9 *Representation of an 8-bit word by a polynomial*



Note that the term is totally omitted if the coefficient is 0, and the coefficient is omitted if it is 1. Also note that x^0 is 1.

Example 4.16

To find the 8-bit word related to the polynomial $x^5 + x^2 + x$, we first supply the omitted terms. Since $n = 8$, it means the polynomial is of degree 7. The expanded polynomial is

$$0x^7 + 0x^6 + 1x^5 + 0x^4 + 0x^3 + 1x^2 + 1x^1 + 0x^0$$

This is related to the 8-bit word 00100110.

Operations

Note that any operation on polynomials actually involves two operations: operations on coefficients and operations on two polynomials. In other words, we need to define two fields: one for the coefficients and one for the polynomials. Coefficients are made of 0 or 1; we can use the **GF**(2) field for this purpose. We discusses this field before (see Example 4.14). For the polynomials we need the field GF(2^n), which we will discuss shortly.

Polynomials representing *n*-bit words use two fields: GF(2) and GF(2^n).

Modulus

Before defining the operations on polynomials, we need to talk about the modulus polynomials. Addition of two polynomials never creates a polynomial out of the set. However, multiplication of two polynomials may create a polynomial with degrees more than $n - 1$. This means we need to divide the result by a modulus and keep only the remainder, as we did in modular arithmetic. For the sets of polynomials in **GF**(2^n), a group of polynomials of degree n is defined as the modulus. The modulus in this case acts as a *prime polynomial,* which means that no polynomials in the set can divide this polynomial. A prime polynomial cannot be factored into a polynomial with degree of less than n. Such polynomials are referred to as ***irreducible*** polynomials. Table 4.4 shows irreducible polynomials of degrees 1 to 5.

For each degree, there is often more than one irreducible polynomial, which means when we define our GF(2^n) we need to declare which irreducible polynomial we are using as the modulus.

Table 4.4 *List of irreducible polynomials*

Degree	*Irreducible Polynomials*
1	$(x + 1)$, (x)
2	$(x^2 + x + 1)$
3	$(x^3 + x^2 + 1)$, $(x^3 + x + 1)$
4	$(x^4 + x^3 + x^2 + x + 1)$, $(x^4 + x^3 + 1)$, $(x^4 + x + 1)$
5	$(x^5 + x^2 + 1)$, $(x^5 + x^3 + x^2 + x + 1)$, $(x^5 + x^4 + x^3 + x + 1)$, $(x^5 + x^4 + x^3 + x^2 + 1)$, $(x^5 + x^4 + x^2 + x + 1)$

Addition

Now let us define the addition operation for polynomials with coefficient in GF(2). Addition is very easy: we add the coefficients of the corresponding terms in GF(2). Note that adding two polynomials of degree $n - 1$ always create a polynomial with degree $n - 1$, which means that we do not need to reduce the result using the modulus.

Example 4.17

Let us do $(x^5 + x^2 + x) \oplus (x^3 + x^2 + 1)$ in GF(2^8). We use the symbol $\oplus$ to show that we mean polynomial addition. The following shows the procedure:

$$
\begin{array}{ll}
0x^7 + 0x^6 + 1x^5 + 0x^4 + 0x^3 + 1x^2 + 1x^1 + 0x^0 & \oplus \\
0x^7 + 0x^6 + 0x^5 + 0x^4 + 1x^3 + 1x^2 + 0x^1 + 1x^0 & \\
\text{---------- (same dashes)} & \\
0x^7 + 0x^6 + 1x^5 + 0x^4 + 1x^3 + 0x^2 + 1x^1 + 1x^0 & \rightarrow \quad x^5 + x^3 + x + 1
\end{array}
$$

There is a short cut: keeps the uncommon terms and delete the common terms. In other words, x^5, x^3, x, and 1 are kept and x^2, which is common in the two polynomials, is deleted.

Example 4.18

There is also another short cut. Because the addition in **GF**(2) means the exclusive-or (XOR) operation. So we can exclusive-or the two words, bits by bits, to get the result. In the previous example, $x^5 + x^2 + x$ is 00100110 and $x^3 + x^2 + 1$ is 00001101. The result is 00101011 or in polynomial notation $x^5 + x^3 + x + 1$.

Additive Identity The additive identity in a polynomial is a zero polynomial (a polynomial with all coefficients set to zero) because adding a polynomial with itself results in a zero polynomial.

Additive Inverse The additive inverse of a polynomial with coefficients in GF(2) is the polynomial itself. This means that the subtraction operation is the same as the addition operation.

Addition and subtraction operations on polynomials are the same operation.

Multiplication

Multiplication in polynomials is the sum of the multiplication of each term of the first polynomial with each term of the second polynomial. However, we need to remember three points. First, the coefficient multiplication is done in GF(2). Second, multiplying x^i by x^j results in x^{i+j}. Third, the multiplication may create terms with degree more than $n - 1$, which means the result needs to be reduced using a modulus polynomial. We first show how to multiply two polynomials according to the above definition. Later we will see a more efficient algorithm that can be used by a computer program.

Example 4.19

Find the result of $(x^5 + x^2 + x) \otimes (x^7 + x^4 + x^3 + x^2 + x)$ in GF(2^8) with irreducible polynomial $(x^8 + x^4 + x^3 + x + 1)$. Note that we use the symbol $\otimes$ to show the multiplication of two polynomials.

Solution

We first multiply the two polynomials as we have learned in algebra. Note that in this process, a pair of terms with equal power of x are deleted. For example, $x^9 + x^9$ is totally deleted because the result is a zero polynomial, as we discussed above.

$$P_1 \otimes P_2 = x^5(x^7 + x^4 + x^3 + x^2 + x) + x^2(x^7 + x^4 + x^3 + x^2 + x) + x(x^7 + x^4 + x^3 + x^2 + x)$$
$$P_1 \otimes P_2 = x^{12} + x^9 + x^8 + x^7 + x^6 + x^9 + x^6 + x^5 + x^4 + x^3 + x^8 + x^5 + x^4 + x^3 + x^2$$
$$P_1 \otimes P_2 = (x^{12} + x^7 + x^2) \bmod (x^8 + x^4 + x^3 + x + 1) = x^5 + x^3 + x^2 + x + 1$$

To find the final result, divide the polynomial of degree 12 by the polynomial of degree 8 (the modulus) and keep only the remainder. The process is the same as we have learned in algebra, but we need to remember that subtraction is the same as addition here. Figure 4.10 shows the process of division.

Figure 4.10 *Polynomial division with coefficients in GF(2)*

$$\begin{array}{r|l} & x^4 + 1 \\ \hline x^8 + x^4 + x^3 + x + 1 & x^{12} + x^7 + x^2 \\ & x^{12} + x^8 + x^7 + x^5 + x^4 \\ \hline & x^8 + x^5 + x^4 + x^2 \\ & x^8 + x^4 + x^3 + x + 1 \\ \hline \text{Remainder} & \boxed{x^5 + x^3 + x^2 + x + 1} \end{array}$$

Multiplicative Identity The multiplicative identity is always 1. For example, in GF(2^8), the multiplicative inverse is the bit pattern 00000001.

Multiplicative Inverse Finding the multiplicative inverse is a little more involved. The extended Euclidean algorithm must be applied to the modulus and the polynomial. The process is exactly the same as for integers.

Example 4.20

In **GF**(2^4), find the inverse of ($x^2 + 1$) modulo ($x^4 + x + 1$).

Solution
We use the extended Euclidean algorithm as in Table 4.5:

Table 4.5 *Euclidean algorithm for Exercise 4.20*

q	r_1	r_2	r	t_1	t_2	t
($x^2 + 1$)	($x^4 + x + 1$)	($x^2 + 1$)	(x)	(0)	(1)	($x^2 + 1$)
(x)	($x^2 + 1$)	(x)	(1)	(1)	($x^2 + 1$)	($x^3 + x + 1$)
(x)	(x)	(1)	(0)	($x^2 + 1$)	($x^3 + x + 1$)	(0)
	(1)	(0)		($x^3 + x + 1$)	(0)	

This means that $(x^2 + 1)^{-1}$ modulo $(x^4 + x + 1)$ is $(x^3 + x + 1)$. The answer can be easily proved by multiplying the two polynomials and finding the remainder when the result is divided by the modulus.

$$[(x^2 + 1) \otimes (x^3 + x + 1)] \bmod (x^4 + x + 1) = 1$$

Example 4.21

In GF(2^8), find the inverse of (x^5) modulo ($x^8 + x^4 + x^3 + x + 1$).

Solution
Use the Extended Euclidean algorithm as shown in Table 4.6:

Table 4.6 *Euclidean algorithm for Example 4.21*

q	r_1	r_2	r	t_1	t_2	t
(x^3)	($x^8 + x^4 + x^3 + x + 1$)	(x^5)	($x^4 + x^3 + x + 1$)	(0)	(1)	(x^3)
($x + 1$)	(x^5)	($x^4 + x^3 + x + 1$)	($x^3 + x^2 + 1$)	(1)	(x^3)	($x^4 + x^3 + 1$)
(x)	($x^4 + x^3 + x + 1$)	($x^3 + x^2 + 1$)	(1)	(x^3)	($x^4 + x^3 + 1$)	($x^5 + x^4 + x^3 + x$)
($x^3 + x^2 + 1$)	($x^3 + x^2 + 1$)	(1)	(0)	($x^4 + x^3 + 1$)	($x^5 + x^4 + x^3 + x$)	(0)
	(1)	(0)		($x^5 + x^4 + x^3 + x$)	(0)	

This means that $(x^5)^{-1}$ modulo $(x^8 + x^4 + x^3 + x + 1)$ is $(x^5 + x^4 + x^3 + x)$. The answer can be easily proved by multiplying the two polynomials and finding the remainder when the result is divided by the modulus.

$$[(x^5) \otimes (x^5 + x^4 + x^3 + x)] \bmod (x^8 + x^4 + x^3 + x + 1) = 1$$

Multiplication Using Computer

Because of the division operation, there is an efficiency problem involved in writing a program to multiply two polynomials. The computer implementation uses a better algorithm, repeatedly multiplying a reduced polynomial by x. For example, instead of finding the result of $(x^2 \otimes P_2)$, the program finds the result of $(x \otimes (x \otimes P_2))$. The benefit of this strategy will be discussed shortly, but first let us use an example to show the process.

Example 4.22

Find the result of multiplying $P_1 = (x^5 + x^2 + x)$ by $P_2 = (x^7 + x^4 + x^3 + x^2 + x)$ in GF(2^8) with irreducible polynomial $(x^8 + x^4 + x^3 + x + 1)$ using the algorithm described above.

Solution

The process is shown in Table 4.7. We first find the partial result of multiplying x^0, x^1, x^2, x^3, x^4, and x^5 by P_2. Note that although only three terms are needed, the product of $x^m \otimes P_2$ for m from 0 to 5 is calculated because each calculation depends on the previous result.

Table 4.7 *An efficient algorithm for multiplication using polynomials (Example 4.22)*

Powers	*Operation*	*New Result*	*Reduction*
$x^0 \otimes P_2$		$x^7 + x^4 + x^3 + x^2 + x$	No
$x^1 \otimes P_2$	$\boldsymbol{x} \otimes (x^7 + x^4 + x^3 + x^2 + x)$	$x^5 + x^2 + x + 1$	**Yes**
$x^2 \otimes P_2$	$\boldsymbol{x} \otimes (x^5 + x^2 + x + 1)$	$x^6 + x^3 + x^2 + x$	No
$x^3 \otimes P_2$	$\boldsymbol{x} \otimes (x^6 + x^3 + x^2 + x)$	$x^7 + x^4 + x^3 + x^2$	No
$x^4 \otimes P_2$	$\boldsymbol{x} \otimes (x^7 + x^4 + x^3 + x^2)$	$x^5 + x + 1$	**Yes**
$x^5 \otimes P_2$	$\boldsymbol{x} \otimes (x^5 + x + 1)$	$x^6 + x^2 + x$	No
$\mathbf{P_1 \times P_2} = (x^6 + x^2 + x) + (x^6 + x^3 + x^2 + x) + (x^5 + x^2 + x + 1) = \boldsymbol{x^5 + x^3 + x^2 + x + 1}$			

The above algorithm has two benefits. First, multiplication of a polynomial by x can be easily achieved by one-bit shifting of the n-bit word; an operation provided by common programming languages. Second, the result needed to be reduced only if the polynomial maximum power is $n - 1$. In this case, reduction can be easily done by an XOR operation with the modulus because the highest power in the result is only 8. We can then design a simple algorithm to find each partial result:

1. If the most significant bit of the previous result is 0, just shift the previous result one bit to the left.
2. If the most significant bit of the previous result is 1,
   a. shift it one bit to the left, and
   b. exclusive-or it with the modulus without the most significant bit

Example 4.23

Repeat Example 4.22 using bit patterns of size 8.

Solution

We have $P_1 = 000100110$, $P_2 = 10011110$, modulus = 100011010 (nine bits). We show the exclusive-or operation by $\oplus$. See Table 4.8.

Table 4.8 *An efficient algorithm for multiplication using n-bit words*

Powers	*Shift-Left Operation*	*Exclusive-Or*
$x^0 \otimes P_2$		10011110
$x^1 \otimes P_2$	00111100	(00111100) $\oplus$ (00011010) = **00100111**
$x^2 \otimes P_2$	01001110	**01001110**
$x^3 \otimes P_2$	10011100	10011100
$x^4 \otimes P_2$	00111000	(00111000) $\oplus$ (00011010) = 00100011
$x^5 \otimes P_2$	01000110	**01000110**
$P_1 \otimes P_2$ = (00100111) $\oplus$ (01001110) $\oplus$ (01000110) = 00101111		

In this case, we need only five shift-left operations and four exclusive-or operations to multiply the two polynomials. In general, a maximum of $n - 1$ shift-left operations and $2n$ exclusive-or operations are needed to multiply two polynomial of degree $n - 1$.

Multiplication of polynomials in GF(2^n) can be achieved using shift-left and exclusive-or operations.

Example 4.24

The GF(2^3) field has 8 elements. We use the irreducible polynomial ($x^3 + x^2 + 1$) and show the addition and multiplication tables for this field. We show both 3-bit words and the polynomials. Note that there are two irreducible polynomials for degree 3. The other one, ($x^3 + x + 1$), yields a totally different table for multiplication. Table 4.9 shows addition. The shaded boxes easily give us the additive inverses pairs.

Table 4.10 shows multiplication. The shaded boxes easily give us the multiplicative inverse pairs.

Using a Generator

Sometimes it is easier to define the elements of the **GF**(2^n) field using a generator. In this field with the irreducible polynomial $f(x)$, an element in the field, a, must satisfy the relation $f(a) = 0$. In particular, if g is a generator of the field, then $f(g) = 0$. It can be proved that the elements of the field can be generated as

$$\{0, g, g, g^2, \ldots, g^N\}, \text{ where } N = 2^n - 2$$

Table 4.9 *Addition table for GF(2^3)*

⊕	000 **(0)**	001 **(1)**	010 **(x)**	011 **(x + 1)**	100 **(x^2)**	101 **$x^2 + 1$**	110 **($x^2 + x$)**	111 **($x^2 + x + 1$)**
000 **(0)**	000 **(0)**	001 **(1)**	010 **(x)**	011 **(x + 1)**	100 **(x^2)**	101 **($x^2 + 1$)**	110 **($x^2 + x$)**	111 **($x^2 + x + 1$)**
001 **(1)**	001 **(1)**	000 **(0)**	011 **(x + 1)**	010 **(x^2)**	101 **($x^2 + 1$)**	100 **($x^2 + x$)**	111 **($x^2 + x + 1$)**	110 **($x^2 + x$)**
010 **(x)**	010 **(x)**	011 **(x + 1)**	000 **(0)**	001 **(1)**	110 **($x^2 + x$)**	111 **($x^2 + x + 1$)**	100 **($x^2 + x$)**	101 **($x^2 + 1$)**
011 **(x + 1)**	011 **(x + 1)**	010 **(x)**	001 **(1)**	000 **(0)**	111 **($x^2 + x + 1$)**	110 **($x^2 + x$)**	101 **($x^2 + 1$)**	100 **(x^2)**
100 **(x^2)**	100 **(x^2)**	101 **($x^2 + 1$)**	110 **($x^2 + x$)**	111 **($x^2 + x + 1$)**	000 **(0)**	001 **(1)**	010 **(x)**	011 **(x + 1)**
101 **($x^2 + 1$)**	101 **($x^2 + 1$)**	100 **(x^2)**	111 **($x^2 + x + 1$)**	110 **($x^2 + x$)**	001 **(1)**	000 **(0)**	011 **(x + 1)**	010 **(x)**
110 **($x^2 + x$)**	110 **($x^2 + x$)**	111 **($x^2 + x + 1$)**	100 **(x^2)**	101 **($x^2 + 1$)**	010 **(x)**	011 **(x + 1)**	000 **(0)**	001 **(1)**
111 **($x^2 + x + 1$)**	111 **($x^2 + x + 1$)**	110 **($x^2 + x$)**	101 **($x^2 + 1$)**	100 **(x^2)**	011 **(x + 1)**	010 **(x)**	001 **(1)**	000 **(0)**

Table 4.10 *Multiplication table for GF(2^3) with irreducible polynomial* ($x^3 + x^2 + 1$)

⊗	000 **(0)**	001 **(1)**	010 **(x)**	011 **(x + 1)**	100 **(x^2)**	101 **($x^2 + 1$)**	110 **($x^2 + x$)**	111 **($x^2 + x + 1$)**
000 (0)	000 (0)	000 (0)	000 (0)	000 (0)	000 (0)	000 (0)	000 (0)	000 (0)
001 (1)	000 (0)	001 (1)	010 **(x)**	011 **(x + 1)**	100 **(x^2)**	101 **($x^2 + 1$)**	110 **($x^2 + x$)**	111 **($x^2 + x + 1$)**
010 **(x)**	000 (0)	010 **(x)**	100 **(x)**	110 **($x^2 + x$)**	101 **($x^2 + 1$)**	111 **($x^2 + x + 1$)**	001 (1)	011 **(x + 1)**
011 **(x + 1)**	000 (0)	011 **(x + 1)**	110 **($x^2 + x$)**	101 **($x^2 + 1$)**	001 (1)	010 **(x)**	111 **($x^2 + x + 1$)**	100 **(x)**
100 **(x^2)**	000 (0)	100 **(x^2)**	101 **($x^2 + 1$)**	001 (1)	111 **($x^2 + x + 1$)**	011 **(x + 1)**	010 **(x)**	110 **($x^2 + x$)**
101 **($x^2 + 1$)**	000 (0)	101 **($x^2 + 1$)**	111 **($x^2 + x + 1$)**	010 **(x)**	011 **(x + 1)**	110 **($x^2 + x$)**	100 **(x^2)**	001 (1)
110 **($x^2 + x$)**	000 (0)	110 **($x^2 + x$)**	001 (1)	111 **($x^2 + x + 1$)**	010 **(x)**	100 **(x^2)**	011 **(x + 1)**	101 **($x^2 + 1$)**
111 **($x^2 + x + 1$)**	000 (0)	111 **($x^2 + x + 1$)**	011 **(x + 1)**	100 **(x^2)**	110 **($x^2 + x$)**	001 (1)	101 **($x^2 + 1$)**	010 **(x)**

Example 4.25

Generate the elements of the field **GF**(2^4) using the irreducible polynomial $f(x) = x^4 + x + 1$.

Solution

The elements 0, g^0, g^1, g^2, and g^3 can be easily generated, because they are the 4-bit representations of 0, 1, x^2, and x^3 (there is no need for polynomial division). Elements g^4 through g^{14}, which represent x^4 though x^{14} need to be divided by the irreducible polynomial. To avoid the polynomial

division, the relation $f(g) = g^4 + g + 1 = 0$ can be used. Using this relation, we have $g^4 = -g - 1$. Because in this field addition and subtraction are the same operation, $g^4 = g + 1$. We use this relation to find the value of all elements as 4-bit words:

0	=	0	=	0	=	0	⟶	0	=	(0000)
g^0	=	g^0	=	g^0	=	g^0	⟶	g^0	=	(0001)
g^1	=	g^1	=	g^1	=	g^1	⟶	g^1	=	(0010)
g^2	=	g^2	=	g^2	=	g^2	⟶	g^2	=	(0100)
g^3	=	g^3	=	g^3	=	g^3	⟶	g^3	=	(1000)
g^4	=	g^4	=	g^4	=	$g + 1$	⟶	g^4	=	(0011)
g^5	=	$g\,(g^4)$	=	$g\,(g + 1)$	=	$g^2 + g$	⟶	g^5	=	(0110)
g^6	=	$g\,(g^5)$	=	$g\,(g^2 + g)$	=	$g^3 + g^2$	⟶	g^6	=	(1100)
g^7	=	$g\,(g^6)$	=	$g\,(g^3 + g)$	=	$g^3 + g + 1$	⟶	g^7	=	(1011)
g^8	=	$g\,(g^7)$	=	$g\,(g^3 + g + 1)$	=	$g^2 + 1$	⟶	g^8	=	(0101)
g^9	=	$g\,(g^8)$	=	$g\,(g^2 + 1)$	=	$g^3 + g$	⟶	g^9	=	(1010)
g^{10}	=	$g\,(g^9)$	=	$g\,(g^3 + g)$	=	$g^2 + g + 1$	⟶	g^{10}	=	(0111)
g^{11}	=	$g\,(g^{10})$	=	$g\,(g^2 + g + 1)$	=	$g^3 + g^2 + g$	⟶	g^{11}	=	(1110)
g^{12}	=	$g\,(g^{11})$	=	$g\,(g^3 + g^2 + g)$	=	$g^3 + g^2 + g + 1$	⟶	g^{12}	=	(1111)
g^{13}	=	$g\,(g^{12})$	=	$g\,(g^3 + g^2 + g + 1)$	=	$g^3 + g^2 + 1$	⟶	g^{13}	=	(1101)
g^{14}	=	$g\,(g^{13})$	=	$g\,(g^3 + g^2 + 1)$	=	$g^3 + 1$	⟶	g^{14}	=	(1001)

The main idea is to reduce terms g^4 to g^{14} to a combination of the terms 1, g, g^2, and g^3, using the relation $g^4 = g + 1$. For example,

$$g^{12} = g\,(g^{11}) = g\,(g^3 + g^2 + g) = g^4 + g^3 + g^2 = g^3 + g^2 + g + 1$$

After the reduction, it is easy to transform the powers into an n-bit word. For example, $g^3 + 1$ is equivalent to 1001, because only the powers 0 and 3 are present. Note that two equal terms cancel each other in this process. For example, $g^2 + g^2 = 0$.

Inverses

Finding inverses using the above representation is simple.

Additive Inverses

The additive inverse of each element is the element itself because addition and subtraction in this field are the same: $-g^3 = g^3$

Multiplicative Inverses

Finding the multiplicative inverse of each element is also very simple. For example, we can find the multiplicative inverse of g^3 as shown below:

$$(g^3)^{-1} = g^{-3} = g^{12} = g^3 + g^2 + g + 1 \quad \rightarrow \quad (1111)$$

Note that the exponents are calculated modulo $2^n - 1$, 15 in this case. Therefore, the exponent $-3 \bmod 15 = 12 \bmod 15$. It can be easily proved that g^3 and g^{12} are inverses of each other because $g^3 \times g^{12} = g^{15} = g^0 = 1$.

Operations

The four operations defined for the field can also be performed using this representation.

Addition and Subtraction

Addition and subtraction are the same operation. The intermediate results can be simplified as shown in the following example.

Example 4.26

The following show the results of addition and subtraction operations:

a. $g^3 + g^{12} + g^7 = g^3 + (g^3 + g^2 + g + 1) + (g^3 + g + 1) = g^3 + g^2 \rightarrow (1100)$

b. $g^3 - g^6 = g^3 + g^6 = g^3 + (g^3 + g^2) = g^2 \rightarrow (0100)$

Multiplication and Division

Multiplication is the addition of powers modulo $2^n - 1$. Division is multiplication using the multiplicative inverse.

Example 4.27

The following show the result of multiplication and division operations:

a. $g^9 \times g^{11} = g^{20} = g^{20 \bmod 15} = g^5 = g^2 + g \rightarrow (0110)$

b. $g^3 / g^8 = g^3 \times g^7 = g^{10} = g^2 + g + 1 \rightarrow (0111)$

Summary

The finite field GF(2^n) can be used to define four operations of addition, subtraction, multiplication and division over n-bit words. The only restriction is that division by zero is not defined. Each n-bit word can also be represented as a polynomial of degree $n - 1$ with coefficients in GF(2), which means that the operations on n-bit words are the same as the operations on this polynomial. To make it modular, we need to define an irreducible polynomial of degree n when we multiply two polynomials. The extended Euclidean algorithm can be applied to polynomials to find the multiplicative inverses.

4.3 RECOMMENDED READING

The following books and Web sites provide more details about subjects discussed in this chapter. The items enclosed in brackets refer to the reference list at the end of the book.

Books

[Dur05], [Ros06], [Bla03], [BW00], and [DF04] discuss algebraic structures thoroughly.

WebSites

The following websites give more information about topics discussed in this chapter.

http://en.wikipedia.org/wiki/Algebraic_structure
http://en.wikipedia.org/wiki/Ring_%28mathematics%29
http://en.wikipedia.org/wiki/Polynomials
http://www.math.niu.edu/~rusin/known-math/index/20-XX.html
http://www.math.niu.edu/~rusin/known-math/index/13-XX.html
http://www.hypermaths.org/quadibloc/math/abaint.htm
http://en.wikipedia.org/wiki/Finite_field