Chapter Four
Finite Fields

4.1. Groups, Rings, and Fields

Groups, rings, and fields are the fundamental elements of a branch of mathematics known as abstract algebra, or modern algebra. In abstract algebra, we are concerned with sets on whose elements we can operate algebraically; that is, we can combine two elements of the set, perhaps in several ways, to obtain a third element of the set. These operations are subject to specific rules, which define the nature of the set. By convention, the notation for the two principal classes of operations on set elements is usually the same as the notation for addition and multiplication on ordinary numbers. However, it is important to note that, in abstract algebra, we are not limited to ordinary arithmetical operations. All this should become clear as we proceed.

4.1.1. Groups

A **group** G, sometimes denoted by $\{G, \cdot\}$ is a set of elements with a binary operation, denoted by $\cdot$, that associates to each ordered pair (a, b) of elements in G an element $(a \cdot b)$ in G, such that the following axioms are obeyed:

Note: The operator $\cdot$ is generic and can refer to addition, multiplication, or some other mathematical operation.

(A1) Closure: If a and b belong to G, then $a \cdot b$ is also in G.

(A2) Associative: $a \cdot (b \cdot c) = (a \cdot b) \cdot c$ for all a, b, c in G.

(A3) Identity element: There is an element e in G such that $a \cdot e = e \cdot a = a$ for all a in G.

(A4) Inverse element: For each a in G there is an element a' in G such that $a \cdot a' = a' \cdot a = e$.

Let N_n denote a set of n distinct symbols that, for convenience, we represent as {1,2,...,n}. A permutation of n distinct symbols is a one-to-one mapping from N_n to N_n. Define S_n to be the set of all permutations of n distinct symbols. Each element of S_n is represented by a permutation of the integers in {1,2,...,n}. It is easy to demonstrate that S_n is a group:

1: If $\alpha, \beta \in S_n$, then the composite mapping $\alpha \cdot \beta$ is formed by permuting the elements of β according to the permutation α. For example, $\{3,2,1\} \cdot \{1,3,2\} = \{2,3,1\}$. Clearly, $\alpha \cdot \beta \in S_n$.

2: The composition of mappings is also easily seen to be associative.

3: The identity mapping is the permutation that does not alter the order of

the n elements. For S_n, the identity element is {1,2,...,n}.

4: For any $\alpha \in S_n$, the mapping that undoes the permutation defined by α is the inverse element for α. There will always be such an inverse. For example $\{2,3,1\} \cdot \{3,1,2\} = \{1,2,3\}$

If a group has a finite number of elements, it is referred to as a **finite group**, and the **order** of the group is equal to the number of elements in the group. Otherwise, the group is an **infinite group**.
A group is said to be abelian if it satisfies the following additional condition:

(A5) Commutative: $a \cdot b = b \cdot a$ for all a, b in G.

The set of integers (positive, negative, and 0) under addition is an abelian group. The set of nonzero real numbers under multiplication is an abelian group. The set S_n from the preceding example is a group but not an abelian group for $n > 2$.

When the group operation is addition, the identity element is 0; the inverse element of a is a; and subtraction is defined with the following rule: a b = a + (b).

Cyclic Group
We define exponentiation within a group as repeated application of the group operator, so that $a^3 = a \cdot a \cdot a$. Further, we define $a^0 = e$, the identity element; and $a^{-n}=(a')^n$. A group G is cyclic if every element of G is a power a^k (k is an integer) of a fixed element $a \in G$. The element a is said to generate the group G, or to be a **generator** of G. A cyclic group is always abelian, and may be finite or infinite.

The additive group of integers is an infinite cyclic group generated by the element 1. In this case, powers are interpreted additively, so that n is the nth power of 1.

4.1.2. Rings

A **ring** R, sometimes denoted by {R, +, x}, is a set of elements with two binary operations, called addition and multiplication, such that for all a, b, c in R the following axioms are obeyed:
[2] Generally, we do not use the multiplication symbol, ×, but denote multiplication by the concatenation of two elements.

(A1-A5) R is an abelian group with respect to addition; that is, R satisfies axioms (A1) through (A5). For the case of an additive group, we denote the identity element as 0 and the inverse of *a* as *a*.

(M1) Closure under multiplication: If *a* and *b* belong to R, then *ab* is also in R.

(M2) Associativity of multiplication: $a(bc) = (ab)c$ for all *a, b, c* in R.

(M3) Distributive laws: $a(b + c) = ab + ac$ for all a, b, c in R.
$(a + b)c = ac + bc$ for all a, b, c in R.

In essence, a ring is a set in which we can do addition, subtraction [a b = a + (-b)], and multiplication without leaving the set.

With respect to addition and multiplication, the set of all n-square matrices over the real numbers is a ring.

A ring is said to be **commutative** if it satisfies the following additional condition:

(M4) Commutativity of multiplication: $ab = ba$ for all a, b in R.

Let S be the set of even integers (positive, negative, and 0) under the usual operations of addition and multiplication. S is a commutative ring. The set of all n-square matrices defined in the preceding example is not a commutative ring.

Next, we define an **integral domain**, which is a commutative ring that obeys the following axioms:

(M5) Multiplicative identity: There is an element 1 in R such that $a1 = 1a = a$ for all a in R.

(M6) No zero divisors: If a, b in R and $ab = 0$, then either $a = 0$ or $b = 0$.

Let S be the set of integers, positive, negative, and 0, under the usual operations of addition and multiplication. S is an integral domain.

4.1.3. Fields

A **field** F, sometimes denoted by {F, +, ×}, is a set of elements with two binary operations, called addition and multiplication, such that for all a, b, c in F the following axioms are obeyed:

((A1M6) F is an integral domain; that is, F satisfies axioms (A1) through (A5) and M1 through M6.

(M7) Multiplicative inverse: For each a in F, except 0, there is an element a^{-1} in F such that $aa^{-1} = (a^{-1})a = 1$.

In essence, a field is a set in which we can do addition, subtraction, multiplication, and division without leaving the set. Division is defined with the following rule: $a/b = a(b^{-1})$.

Familiar examples of fields are the rational numbers, the real numbers, and the complex numbers. Note that the set of all integers is not a field, because not every element of the set has a multiplicative inverse; in fact, only the elements 1

and -1 have multiplicative inverses in the integers.

Figure 4.1 summarizes the axioms that define groups, rings, and fields.

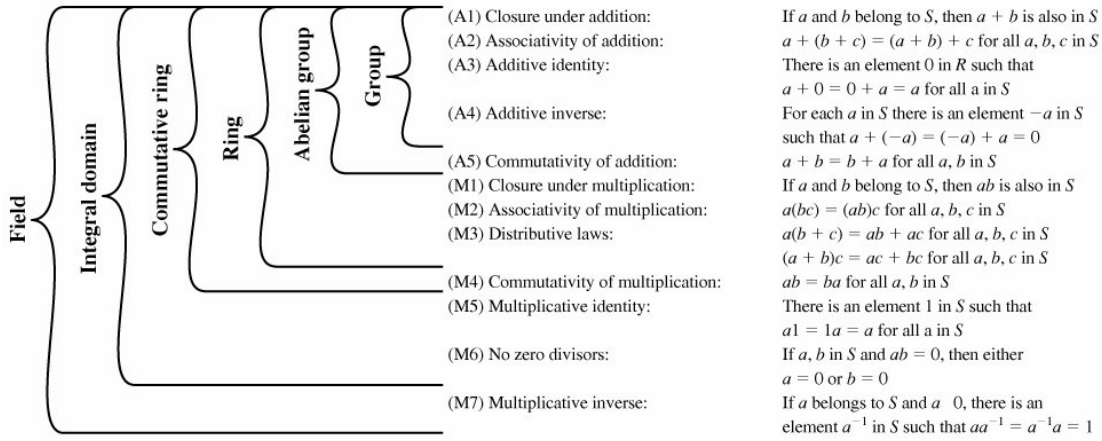


Figure 4.1. Group, Ring, and Field

4.2. Modular Arithmetic

Given any positive integer n and any nonnegative integer *a*, if we divide a by *n*, we get an integer quotient *q* and an integer remainder *r* that obey the following relationship:

Equation 4-1

$$a = qn + r \qquad 0 \leq r < n; q = \lfloor a/n \rfloor$$

where $\lfloor x \rfloor$ is the largest integer less than or equal to *x*.

Figure 4.2 demonstrates that, given a and positive *n*, it is always possible to find *q* and *r* that satisfy the preceding relationship. Represent the integers on the number line; a will fall somewhere on that line (positive a is shown, a similar demonstration can be made for negative a). Starting at 0, proceed to *n*, 2n, up to *qn* such that $qn \leq a$ and $(q + 1)n > a$. The distance from *qn* to *a* is *r*, and we have found the unique values of *q* and *r*. The remainder *r* is often referred to as a **residue**.

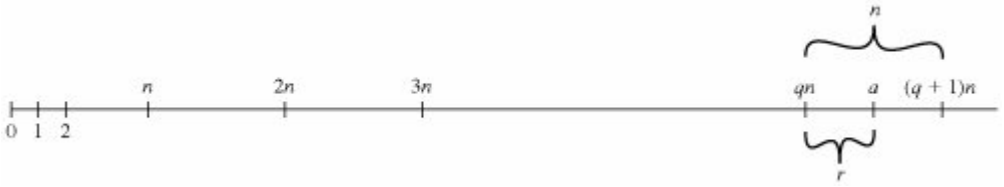


Figure 4.2. The Relationship *a* = *qn* + *r*, 0 *r* < *n*

a = 11;	*n* = 7;	11 = 1 × 7 + 4;	r = 4	*q* = 1
a = -11;	*n* = 7;	-11 = (-2) × 7 + 3;	r = 3	*Q* = -2

If a is an integer and n is a positive integer, we define a mod n to be the remainder when a is divided by n. The integer n is called the **modulus**. Thus, for any integer a, we can always write:

$a = \lfloor a/n \rfloor \times n + (a \bmod n)$

11 mod 7 = 4;	-11 mod 7 = 3

Two integers *a* and *b* are said to be congruent modulo *n*, if $(a \bmod n) = (b \bmod n)$. This is written as $a \equiv b \pmod{n}$.

We have just used the operator mod in two different ways: first as a binary operator that produces a remainder, as in the expression a mod b; second as a congruence relation that shows the equivalence of two integers, as in the expression To distinguish the two uses, the mod term is enclosed in parentheses for a congruence relation; this is common but not universal in the literature.

$73 \equiv 4 \pmod{23}$;	$21 \equiv -9 \pmod{10}$

4.2.1. Divisors

We say that *a* nonzero *b* divides *a* if $a = mb$ for some *m*, where *a*, *b*, and *m* are integers. That is, *b* divides *a* if there is no remainder on division. The notation is commonly used to mean *b* divides *a*. Also, if $b|a$, we say that *b* is *a* **divisor** of *a*.

The positive divisors of 24 are 1, 2, 3, 4, 6, 8, 12, and 24.

The following relations hold:

- If $a|1$, then $a = \pm 1$.
- If $a|b$ and $b|a$, then $a = \pm b$.
- Any $b \neq 0$ divides 0.
- If $b|g$ and $b|h$, then $b|(mg + nh)$ for arbitrary integers *m* and *n*.

To see this last point, note that

If $b|g$, then *g* is of the form $g = b \times g_1$ for some integers g_1.

If $b|h$, then *h* is of the form $h = b \times h_1$ for some integers h_1.

So

$mg + nh = mbg_1 + nbh_1 = b \times (mg_1 + nh_1)$

and therefore *b* divides $mg + nh$.

$b = 7$; $g = 14$; $h = 63$; $m = 3$; $n = 2$. $7|14$ and $7|63$. To show: $7|(3 \times 14 + 2 \times 63)$ We have $(3 \times 14 + 2 \times 63) = 7(3 \times 2 + 2 \times 9)$ And it is obvious that $7|(7(3 \times 2 + 2 \times 9))$

Note that if $a \equiv 0 \pmod{n}$, then $n|a$.

4.2.2. Properties of Congruences

Congruences have the following properties:

1. $a \equiv b \pmod{n}$ if $n|(a\ b)$.
2. $a \equiv b \pmod{n}$ implies $b \equiv a \pmod{n}$..
3. $a \equiv b \pmod{n}$ and $b \equiv c \pmod{n}$ imply $a \equiv c \pmod{n}$.

To demonstrate the first point, if $n|(a\ b)$, then $(a\ b) = kn$ for some k. So we can write $a = b + kn$. Therefore, (a mod n) = (reminder when b + kn is divided by n) = (reminder when b is divided by n) = (b mod n)

$23 \equiv 8 \pmod{5}$	because	$23 - 8 = 15 = 5 \times 3$
$11 \equiv 5 \pmod{8}$	because	$11 - 5 = 16 = 8 \times (2)$
$81 \equiv 0 \pmod{27}$	because	$81 - 0 = 81 = 27 \times 3$

The remaining points are as easily proved.

4.2.3. Modular Arithmetic Operations

Note that, by definition (Figure 4.2), the (mod n) operator maps all integers into the set of integers {0, 1,... (n 1)}. This suggests the question: Can we perform arithmetic operations within the confines of this set? It turns out that we can; this technique is known as **modular arithmetic**.

Modular arithmetic exhibits the following properties:

1. $[(a \bmod n) + (b \bmod n)] \bmod n = (a + b) \bmod n$
2. $[(a \bmod n)\ (b \bmod n)] \bmod n = (a\ b) \bmod n$
3. $[(a \bmod n) \times (b \bmod n)] \bmod n = (a \times b) \bmod n$

We demonstrate the first property. Define $(a \bmod n) = r_a$ and $(b \bmod n) = r_b$. Then we can write $a = r_a + jn$ for some integer j and $b = r_b + kn$ for some integer k. Then

$$\begin{aligned}(a + b) \bmod n &= (r_a + jn + r_b + kn) \bmod n \\ &= (r_a + r_b\ (k + j)n) \bmod n \\ &= (r_a + r_b) \bmod n \\ &= [(a \bmod n] + (b \bmod n)] \bmod n\end{aligned}$$

The remaining properties are as easily proved. Here are examples of the three properties:

$11 \bmod 8 = 3;\ 15 \bmod 8 = 7$
$[(11 \bmod 8) + (15 \bmod 8)] \bmod 8 = 10 \bmod 8 = 2$ $(11 + 15) \bmod 8 = 26 \bmod 8 = 2$
$[(11 \bmod 8)\ (15 \bmod 8)] \bmod 8 = 4 \bmod 8 = 4$ $(11\ 15) \bmod 8 = 4 \bmod 8 = 4$
$[(11 \bmod 8) \times (15 \bmod 8)] \bmod 8 = 21 \bmod 8 = 5$ $(11 \times 15) \bmod 8 = 165 \bmod 8 = 5$

Exponentiation is performed by repeated multiplication, as in ordinary arithmetic.

To find 11^7 mod 13, we can proceed as follows:
$11^2 = 121 \equiv 4 \pmod{13}$
$11^4 = (11^2)^2 \equiv 4^2 \equiv 3 \pmod{13}$
$11^7 \equiv 11 \times 4 \times 3 \equiv 132 \equiv 2 \pmod{13}$

Thus, the rules for ordinary arithmetic involving addition, subtraction, and multiplication carry over into modular arithmetic.

Table 4.1. Arithmetic Modulo 8

+	0	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7	0
2	2	3	4	5	6	7	0	1
3	3	4	5	6	7	0	1	2
4	4	5	6	7	0	1	2	3
5	5	6	7	0	1	2	3	4
6	6	7	0	1	2	3	4	5
7	7	0	1	2	3	4	5	6

(a) Addition modulo 8

×	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7
2	0	2	4	6	0	2	4	6
3	0	3	6	1	4	7	2	5
4	0	4	0	4	0	4	0	4
5	0	5	2	7	4	1	6	3
6	0	6	4	2	0	6	4	2
7	0	7	6	5	4	3	2	1

(b) Multiplication modulo 8

w	$-w$	w^{-1}
0	0	—
1	7	1
2	6	—
3	5	3
4	4	—
5	3	5
6	2	—
7	1	7

(c) Additive and multiplicative inverses modulo 8

Table 4.1 provides an illustration of modular addition and multiplication modulo 8. Looking at addition, the results are straightforward and there is a regular pattern to the matrix. Both matrices are symmetric about the main diagonal, in conformance to the commutative property of addition and multiplication. As in ordinary addition, there is an additive inverse, or negative, to each integer in modular arithmetic. In this case, the negative of an integer x is the integer y such that (x + y) mod 8 = 0. To find the additive inverse of an integer in the left-hand column, scan across the corresponding row of the matrix to find the value 0; the integer at the top of that column is the additive inverse; thus (2 + 6) mod 8 = 0. Similarly, the entries in the

multiplication table are straightforward. In ordinary arithmetic, there is a multiplicative inverse, or reciprocal, to each integer. In modular arithmetic mod 8, the multiplicative inverse of x is the integer y such that (x x y) mod 8 = 1 mod 8. Now, to find the multiplicative inverse of an integer from the multiplication table, scan across the matrix in the row for that integer to find the value 1; the integer at the top of that column is the multiplicative inverse; thus (3 x 3) mod 8 = 1. Note that not all integers mod 8 have a multiplicative inverse; more about that later.

4.2.4. Properties of Modular Arithmetic

Define the set Z_n as the set of nonnegative integers less than n:

$$Z_n = \{0, 1,...,(n\ 1)\}$$

This is referred to as the set of residues, or **residue classes** modulo n. To be more precise, each integer in Z_n represents a residue class. We can label the residue classes modulo n as [0], [1], [2],...,[n 1], where
[r] = {a: a is an integer, a ≡r (mod n)}

The residue classes modulo 4 are	
	[0] = { ..., 16, 12, 8, 4, 0, 4, 8, 12, 16,... }
	[1] = { ..., 15, 11, 7, 3, 1, 5, 9, 13, 17,... }
	[2] = { ..., 14, 10, 6, 2, 2, 6, 10, 14, 18,... }
	[3] = { ..., 13, 9, 5, 1, 3, 7, 11, 15, 19,... }

Of all the integers in a residue class, the smallest nonnegative integer is the one usually used to represent the residue class. Finding the smallest nonnegative integer to which k is congruent modulo n is called reducing k modulo n.
If we perform modular arithmetic within Z_n, the properties shown in Table 4.2 hold for integers in Z_n. Thus, Z_n is a commutative ring with a multiplicative identity element (Figure 4.1).

There is one peculiarity of modular arithmetic that sets it apart from ordinary arithmetic. First, observe that, as in ordinary arithmetic, we can write the following:
Equation 4-2

$$\textbf{if } (a + b) \equiv (a + c) \pmod{n} \quad \textbf{then} \quad b \equiv c \pmod{n}$$

(5 + 23) ≡ (5 + 7)(mod 8}; 23 ≡7 (mod 8)

Equation (4.2) is consistent with the existence of an additive inverse. Adding the additive inverse of a to both sides of Equation (4.2), we have:
((a) + a + b) ≡ ((a) + a + c)(mod n)

$b \equiv c \pmod{n}$
However, the following statement is true only with the attached condition:
Equation 4-3

$$\textbf{if}(a \times b) \equiv (a \times c) \pmod{n} \ \textbf{then}\ b \equiv c \pmod{n} \quad \textbf{if}\ a \text{ is relatively prime to } n$$

Table 4.2. Properties of Modular Arithmetic for Integers in Z_n

Property	Expression
Commutative laws	$(w + x) \bmod n = (x + w) \bmod n$ $(w \times x) \bmod n = (x \times w) \bmod n$
Associative laws	$[(w + x) + y] \bmod n = [w + (x + y)] \bmod n$ $[(w \times x) \times y] \bmod n = [w \times (x \times y)] \bmod n$
Distributive laws	$[w + (x + y)] \bmod n = [(w \times x) + (w \times y)] \bmod n$ $[w + (x \times y)] \bmod n = [(w + x) \times (w + y)] \bmod n$
Identities	$(0 + w) \bmod n = w \bmod n$ $(1 + w) \bmod n = w \bmod n$
Additive inverse (-w)	For each $w \in Z_n$, there exists a z such that $w + z \equiv 0 \bmod n$

where the term relatively prime is defined as follows: two integers are **relatively prime** if their only common positive integer factor is 1. Similar to the case of Equation (4.2), we can say that Equation (4.3) is consistent with the existence of a multiplicative inverse. Applying the multiplicative inverse of a to both sides of Equation (4.2), we have:
$((a^1)ab) \equiv ((a^1)ac) \pmod{n}$
$b \equiv c \pmod{n}$

To see this, consider an example in which the condition of Equation (4.3) does not hold. The integers 6 and 8 are not relatively prime, since they have the common factor 2. We have the following:
$6 \times 3 = 18 \equiv 2 \pmod{8}$
$6 \times 7 = 42 \equiv 2 \pmod{8}$
Yet $3 \neq 7 \pmod{8}$.

The reason for this strange result is that for any general modulus n, a multiplier a that is applied in turn to the integers 0 through (n 1) will fail to produce a complete set of residues if a and n have any factors in common.

With a = 6 and n = 8,

Z_8	0	1	2	3	4	5	6	7
Multiply by 6	0	6	12	18	24	30	36	42
Residues	0	6	4	2	0	6	4	2

Because we do not have a complete set of residues when multiplying by 6, more than one integer in Z_8 maps into the same residue. Specifically, $6 \times 0 \bmod 8 = 6 \times 4 \bmod 8$; $6 \times 1 \bmod 8 = 6 \times 5 \bmod 8$; and so on. Because this is a many-to-one mapping, there is not a unique inverse to the multiply operation.

However, if we take a = 5 and n = 8, whose only common factor is 1,

Z_8	0	1	2	3	4	5	6	7
Multiply by 6	0	5	10	15	20	25	30	35
Residues	0	5	2	7	4	1	6	3

The line of residues contains all the integers in Z_8, in a different order.

In general, an integer has a multiplicative inverse in Z_n if that integer is relatively prime to n. Table 4.1c shows that the integers 1, 3, 5, and 7 have a multiplicative inverse in Z_8, but 2, 4, and 6 do not.

4.3. The Euclidean Algorithm

One of the basic techniques of number theory is the Euclidean algorithm, which is a simple procedure for determining the greatest common divisor of two positive integers.

4.3.1. Greatest Common Divisor

Recall that nonzero b is defined to be a divisor of a if a = mb for some m, where a, b, and m are integers. We will use the notation gcd(a, b) to mean the **greatest common divisor** of a and b. The positive integer c is said to be the greatest common divisor of a and b if

1. c is a divisor of a and of b;
2. any divisor of a and b is a divisor of c.

An equivalent definition is the following:

gcd(a, b) = max[k, such that k|a and k|b]

Because we require that the greatest common divisor be positive, gcd(a, b) = gcd(a, b) = gcd(a, b) = gcd(a, b). In general, gcd(a, b) = gcd(|a|, |b|).

gcd(60, 24) = gcd(60, 24) = 12

Also, because all nonzero integers divide 0, we have gcd(a, 0) = |a|.

We stated that two integers a and b are relatively prime if their only common positive integer factor is 1. This is equivalent to saying that a and b are relatively prime if gcd(a, b) = 1.

8 and 15 are relatively prime because the positive divisors of 8 are 1, 2, 4, and 8, and the positive divisors of 15 are 1, 3, 5, and 15, so 1 is the only integer on both lists.

4.3.2. Finding the Greatest Common Divisor

The Euclidean algorithm is based on the following theorem: For any nonnegative integer a and any positive integer b,

Equation 4-4

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

gcd(55, 22) = gcd(22, 55 mod 22) = gcd(22, 11) = 11

To see that Equation (4.4) works, let d = gcd(a, b). Then, by the definition of gcd, d|a and d|b. For any positive integer b, a can be expressed in the form

$$a = kb + r \equiv r \pmod{b}$$

$$a \bmod b = r$$

with k, r integers. Therefore, (a mod b) = a kb for some integer k. But because d|b, it also divides kb. We also have d|a. Therefore, d|(a mod b). This shows that d is a common divisor of b and (a mod b). Conversely, if d is a common divisor of b and (a mod b), then d|kb and thus d|[kb + (a mod b)], which is equivalent to d|a. Thus, the set of common divisors of a and b is equal to the set of common divisors of b and (a mod b). Therefore, the gcd of one pair is the same as the gcd of the other pair, proving the theorem.

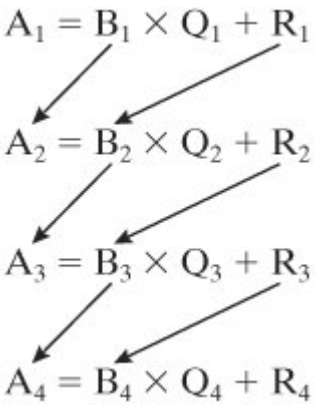
Equation (4.4) can be used repetitively to determine the greatest common divisor.

gcd(18, 12) = gcd(12, 6) = gcd(6, 0) = 6
gcd(11, 10) = gcd(10, 1) = gcd(1, 0) = 1

The Euclidean algorithm makes repeated use of Equation (4.4) to determine the greatest common divisor, as follows. The algorithm assumes a > b > 0. It is acceptable to restrict the algorithm to positive integers because gcd(a, b) = gcd(|a|, |b|).

EUCLID(a, b)
1. A ← a; B ← b
2. if B = 0 return A = gcd(a, b)
3. R = A mod B
4. A ← B
5. B ← R
6. goto 2

The algorithm has the following progression:



To find gcd(1970, 1066)		
1970	= 1 x 1066 + 904	gcd(1066, 904)
1066	= 1 x 904 + 162	gcd(904, 162)
904	= 5 x 162 + 94	gcd(162, 94)
162	= 1 x 94 + 68	gcd(94, 68)
94	= 1 x 68 + 26	gcd(68, 26)
68	= 2 x 26 + 16	gcd(26, 16)
26	= 1 x 16 + 10	gcd(16, 10)
16	= 1 x 10 + 6	gcd(10, 6)
10	= 1 x 6 + 4	gcd(6, 4)
6	= 1 x 4 + 2	gcd(4, 2)
4	= 2 x 2 + 0	gcd(2, 0)
Therefore, gcd(1970, 1066) = 2		

The alert reader may ask how we can be sure that this process terminates. That is, how can we be sure that at some point B divides A? If not, we would get an endless sequence of positive integers, each one strictly smaller than the one before, and this is clearly impossible.

4.4. Finite Fields of The Form GF(p)

In Section 4.1, we defined a field as a set that obeys all of the axioms of Figure 4.1 and gave some examples of infinite fields. Infinite fields are not of particular interest in the context of cryptography. However, finite fields play a crucial role in many cryptographic algorithms. It can be shown that the order of a finite field (number of elements in the field) must be a power of a prime p^n, where n is a positive integer. Here, we need only say that a prime number is an integer whose only positive integer

factors are itself and 1. That is, the only positive integers that are divisors of p are p and 1.

The finite field of order p^n is generally written GF(p^n); stands for Galois field, in honor of the mathematician who first studied finite fields. Two special cases are of interest for our purposes. For n = 1, we have the finite field GF(p); this finite field has a different structure than that for finite fields with n > 1 and is studied in this section. In Section 4.6, we look at finite fields of the form GF(2^n).

4.4.1. Finite Fields of Order p

For a given prime, p, the finite field of order p, GF(p) is defined as the set Z_p of integers {0, 1,..., p 1}, together with the arithmetic operations modulo p.

Recall that we showed in Section 4.2 that the set Z_n of integers {0,1,...,n 1}, together with the arithmetic operations modulo n, is a commutative ring (Table 4.2). We further observed that any integer in Z_n has a multiplicative inverse if and only if that integer is relatively prime to n [see discussion of Equation (4.3)].[4] If n is prime, then all of the nonzero integers in Z_n are relatively prime to n, and therefore there exists a multiplicative inverse for all of the nonzero integers in Z_n. Thus, we can add the following properties to those listed in Table 4.2 for Z_p:

[4] As stated in the discussion of Equation (4.3), two integers are relatively prime if their only common positive integer factor is 1.

Multiplicative inverse (w^1)	For each $w \in Z_p$, $w \neq 0$, there exists a $z \in Z_p$ such that $w \times z \equiv 1 \pmod p$

Because w is relatively prime to p, if we multiply all the elements of Z_p by w, the resulting residues are all of the elements of Z_p permuted. Thus, exactly one of the residues has the value 1. Therefore, there is some integer Z_p in that, when multiplied by w, yields the residue 1. That integer is the multiplicative inverse of w, designated w^1. Therefore, Z_p is in fact a finite field. Further, Equation (4.3) is consistent with the existence of a multiplicative inverse and can be rewritten without the condition:

Equation 4-5

$$\textbf{if } (a \times b) \equiv (a \times c) \pmod p \textbf{ then } b \equiv c \pmod p$$

Multiplying both sides of Equation (4.5) by the multiplicative inverse of a, we have:

$$((a^1) \times a \times b) \equiv ((a^1) \times a \times c) \pmod p$$

$$b \equiv c \pmod p$$

The simplest finite field is GF(2). Its arithmetic operations are easily summarized:

+	0	1
0	0	1
1	1	0

Addition

×	0	1
0	0	0
1	0	1

Multiplication

w	$-w$	w^{-1}
0	0	—
1	1	1

Inverses

In this case, addition is equivalent to the exclusive-OR (XOR) operation, and multiplication is equivalent to the logical AND operation.

Table 4.3 shows GF(7). This is a field of order 7 using modular arithmetic modulo 7. As can be seen, it satisfies all of the properties required of a field (Figure 4.1). Compare this table with Table 4.1. In the latter case, we see that the set Z_8 using modular arithmetic modulo 8, is not a field. Later in this chapter, we show how to define addition and multiplication operations on Z_8 in such a way as to form a finite field.

Table 4.3. Arithmetic in GF(7)

+	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	2	3	4	5	6	0
2	2	3	4	5	6	0	1
3	3	4	5	6	0	1	2
4	4	5	6	0	1	2	3
5	5	6	0	1	2	3	4
6	6	0	1	2	3	4	5

(a) Addition modulo 7

×	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6
2	0	2	4	6	1	3	5
3	0	3	6	2	5	1	4
4	0	4	1	5	2	6	3
5	0	5	3	1	6	4	2
6	0	6	5	4	3	2	1

(b) Multiplication modulo 7

w	$-w$	w^{-1}
0	0	—
1	6	1
2	5	4
3	4	5
4	3	2
5	2	3
6	1	6

(c) Additive and multiplicative inverses modulo 7

4.4.2. Finding the Multiplicative Inverse in GF(p)

It is easy to find the multiplicative inverse of an element in GF(p) for small values of p. You simply construct a multiplication table, such as shown in Table 4.3b, and the desired result can be read directly. However, for large values of p, this approach is not practical.

If gcd(m, b) = 1, then b has a multiplicative inverse modulo m. That is, for positive integer b < m, there exists a b^1 < m such that $bb^1 = 1 \bmod m$. The Euclidean algorithm can be extended so that, in addition to finding gcd(m, b), if the gcd is 1, the algorithm returns the multiplicative inverse of b.

EXTENDED EUCLID(m, b)

1. (A1, A2, A3) ← (1, 0, m); (B1, B2, B3) ← (0, 1, b)
2. if B3 = 0 return A3 = gcd(m, b); no inverse
3. if B3 = 1 return B3 = gcd(m, b); B2 = $b^1 \bmod m$
4. $Q = \left\lfloor \frac{A3}{B3} \right\rfloor$

5. (T1, T2, T3) ← (A1 − QB1, A2 − QB2, A3 − QB3)
6. (A1, A2, A3) ← (B1, B2, B3)
7. (B1, B2, B3) ← (T1, T2, T3)
8. goto 2

Throughout the computation, the following relationships hold:

$$mT1 + bT2 = T3 \quad mA1 + bA2 = A3 \quad mB1 + bB2 = B3$$

To see that this algorithm correctly returns gcd(m, b), note that if we equate A and B in the Euclidean algorithm with A3 and B3 in the extended Euclidean algorithm, then the treatment of the two variables is identical. At each iteration of the Euclidean algorithm, A is set equal to the previous value of B and B is set equal to the previous value of A mod B. Similarly, at each step of the extended Euclidean algorithm, A3 is set equal to the previous value of B3, and B3 is set equal to the previous value of A3 minus the integer quotient of A3 multiplied by B3. This latter value is simply the remainder of A3 divided by B3, which is A3 mod B3.
Note also that if gcd(m, b) = 1, then on the final step we would have B3 = 0 and A3 = 1. Therefore, on the preceding step, B3 = 1. But if B3 = 1, then we can say the following:

$$mB1 + bB2 = B3$$
$$mB1 + bB2 = 1$$
$$bB2 = 1 - mB1$$
$$bB2 \equiv 1 \pmod{m}$$

And B2 is the multiplicative inverse of b, modulo m.

Table 4.4 is an example of the execution of the algorithm. It shows that gcd(1759, 550) = 1 and that the multiplicative inverse of 550 is 355; that is, $550 \times 335 \equiv 1 \pmod{1759}$.

Table 4.4. Finding the Multiplicative Inverse of 550 in GF(1759)

Q	A1	A2	A3	B1	B2	B3
	1	0	1759	0	1	550
3	0	1	550	1	3	109
5	1	3	109	5	16	5
21	5	16	5	106	339	4
1	106	339	4	111	355	1

Summary
In this section, we have shown how to construct a finite field of order p, where p is prime. Specifically, we defined GF(p) with the following properties:

1. GF(p) consists of p elements.

2. The binary operations + and x are defined over the set. The operations of addition, subtraction, multiplication, and division can be performed without leaving the set. Each element of the set other than 0 has a multiplicative inverse.

We have shown that the elements of GF(p) are the integers {0, 1,..., p} and that the arithmetic operations are addition and multiplication mod p.

4.5. Polynomial Arithmetic

Before pursuing our discussion of finite fields, we need to introduce the interesting subject of polynomial arithmetic. We are concerned with polynomials in a single variable x, and we can distinguish three classes of polynomial arithmetic:

- Ordinary polynomial arithmetic, using the basic rules of algebra
- Polynomial arithmetic in which the arithmetic on the coefficients is performed modulo p; that is, the coefficients are in GF(p)
- Polynomial arithmetic in which the coefficients are in GF(p), and the polynomials are defined modulo a polynomial m(x) whose highest power is some integer n

This section examines the first two classes, and the next section covers the last class.

4.5.1. Ordinary Polynomial Arithmetic

A **polynomial** of degree n (integer $n \geq 0$) is an expression of the form

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0 = \sum_{i=0}^{n} a_i x^i$$

where the a_i are elements of some designated set of numbers S, called the **coefficient set**, and $a_n \neq 0$. We say that such polynomials are defined over the coefficient set S.

A zeroth-degree polynomial is called a constant polynomial and is simply an element of the set of coefficients. An nth-degree polynomial is said to be a **monic polynomial** if $a_n = 1$.

In the context of abstract algebra, we are usually not interested in evaluating a polynomial for a particular value of x [e.g., f(7)]. To emphasize this point, the variable x is sometimes referred to as the indeterminate.

Polynomial arithmetic includes the operations of addition, subtraction, and multiplication. These operations are defined in a natural way as though the variable x was an element of S. Division is similarly defined, but requires that S be a field. Examples of fields include the real numbers, rational numbers, and Z_p for p prime. Note that the set of all integers is not a field and does not support polynomial division.

Addition and subtraction are performed by adding or subtracting corresponding coefficients. Thus, if

$$f(x) = \sum_{i=0}^{n} a_i x^i; \quad g(x) = \sum_{i=0}^{m} b_i x^i; \quad n \geq m$$

then addition is defined as

$$f(x) + g(x) = \sum_{i=0}^{m} (a_i + b_i) x^i + \sum_{i=m+1}^{n} a_i x^i$$

and multiplication is defined as

$$f(x) \times g(x) = \sum_{i=0}^{n+m} c_i x^i$$

where

$$c_k = a_0 b_{k1} + a_1 b_{k1} + ... + a_{k1} b_1 + a_k b_0$$

In the last formula, we treat a_i as zero for i > n and b_i as zero for i > m. Note that the degree of the product is equal to the sum of the degrees of the two polynomials.

As an example, let $f(x) = x^3 + x^2 + 2$ and $g(x) = x^2\ x + 1$, where S is the set of integers. Then

$f(x) + g(x) = x^3 + 2x^2\ x + 3$

$f(x)\ g(x) = x^3 + x + 1$

$f(x)\ x\ g(x) = x^5 + 3x^2\ 2x + 2$

Figures 4.3a through 4.3c show the manual calculations. We comment on division subsequently.

$$\begin{array}{r} x^3 + x^2 \quad + 2 \\ +\ (x^2 - x + 1) \\ \hline x^3 + 2x^2 - x + 3 \end{array}$$

(a) Addition

$$\begin{array}{r} x^3 + x^2 \quad + 2 \\ -\ (x^2 - x + 1) \\ \hline x^3 \quad + x + 1 \end{array}$$

(b) Subtraction

$$\begin{array}{r} x^3 + x^2 \quad + 2 \\ \times\ (x^2 - x + 1) \\ \hline x^3 + x^2 \quad + 2 \\ -x^4 - x^3 \quad - 2x \\ x^5 + x^4 \quad + 2x^2 \\ \hline x^5 \quad + 3x^2 - 2x + 2 \end{array}$$

(c) Multiplication

$$\begin{array}{r|l} & x + 2 \\ \hline x^2 - x + 1 & x^3 + x^2 \quad + 2 \\ & x^3 - x^2 + x \\ \hline & 2x^2 - x + 2 \\ & 2x^2 - 2x + 2 \\ \hline & x \end{array}$$

(d) Division

Figure 4.3. Examples of Polynomial Arithmetic

4.5.2. Polynomial Arithmetic with Coefficients in Zp

Let us now consider polynomials in which the coefficients are elements of some field F. We refer to this as a polynomial over the field F. In that case, it is easy to show that the set of such polynomials is a ring, referred to as a **polynomial ring**. That is, if we consider each distinct polynomial to be an element of the set, then that set is a ring.

In fact, the set of polynomials whose coefficients are elements of a commutative ring forms a polynomial ring, but that is of no interest in the present context.

When polynomial arithmetic is performed on polynomials over a field, then division is possible. Note that this does not mean that exact division is possible. Let us clarify this distinction. Within a field, given two elements a and b, the quotient a/b is also an element of the field. However, given a ring R that is not a field, in general division will result in both a quotient and a remainder; this is not exact division.

Consider the division 5/3 within a set S. If S is the set of rational numbers, which is a field, then the result is simply expressed as 5/3 and is an element of S. Now suppose that S is the field Z_7. In this case, we calculate (using Table 4.3c):

$$5/3 = (5 \times 3^{1}) \bmod 7 = (5 \times 5) \bmod 7 = 4$$

which is an exact solution. Finally, suppose that S is the set of integers, which is a ring but not a field. Then 5/3 produces a quotient of 1 and a remainder of 2:

$$5/3 = 1 + 2/3$$
$$5 = 1 \times 3 + 2$$

Thus, division is not exact over the set of integers.

Now, if we attempt to perform polynomial division over a coefficient set that is not a field, we find that division is not always defined.

If the coefficient set is the integers, then $(5x^2)/(3x)$ does not have a solution, because it would require a coefficient with a value of 5/3, which is not in the coefficient set. Suppose that we perform the same polynomial division over Z_7. Then we have $(5x^2)/(3x) = 4x$ which is a valid polynomial over Z_7.

However, as we demonstrate presently, even if the coefficient set is a field, polynomial division is not necessarily exact. In general, division will produce a quotient and a remainder:

Equation 4-6

$$\frac{f(x)}{g(x)} = q(x) + \frac{r(x)}{g(x)}$$
$$f(x) = q(x)g(x) + r(x)$$

If the degree of f(x) is n and the degree of g(x) is m, (m n), then the degree of the quotient q(x) m n is and the degree of the remainder is at most m - 1. With the understanding that remainders are allowed, we can say that polynomial division is possible if the coefficient set is a field.

In an analogy to integer arithmetic, we can write f(x) mod g(x) for the remainder r(x) in Equation (4.6). That is, r(x) = f(x) mod g(x). If there is no remainder [i.e., r(x) = 0], then we can say g(x) divides f(x), written as g(x)|f(x); equivalently, we can say that g(x) is a factor of f(x) or g(x) is a divisor of f(x).

For the preceding example and [$f(x) = x^3 + x^2 + 2$ and $g(x) = x^2 x + 1$], f(x)/g(x) produces a quotient of q(x) = x + 2 and a remainder r(x) = x as shown in Figure 4.3d. This is easily verified by noting that

$q(x)g(x) + r(x) = (x + 2)(x^2 x + 1) + x = (x^3 + x^2 x + 2) + x$
$= x^3 + x^2 + 2 = f(x)$

For our purposes, polynomials over GF(2) are of most interest. Recall from Section 4.4 that in GF(2), addition is equivalent to the XOR operation, and multiplication is

equivalent to the logical AND operation. Further, addition and subtraction are equivalent mod 2: 1 + 1 = 1 1 = 0; 1 + 0 = 1 0 = 1; 0 + 1 = 0 1 = 1.

Figure 4.4 shows an example of polynomial arithmetic over GF(2). For f(x) = ($x^7 + x^5 + x^4 + x^3 + x + 1$) and g(x) = ($x^3 + x + 1$), the figure shows f(x) + g(x); f(x) g(x); f(x) x g(x); and f(x)/g(x). Note that g(x)|f(x)

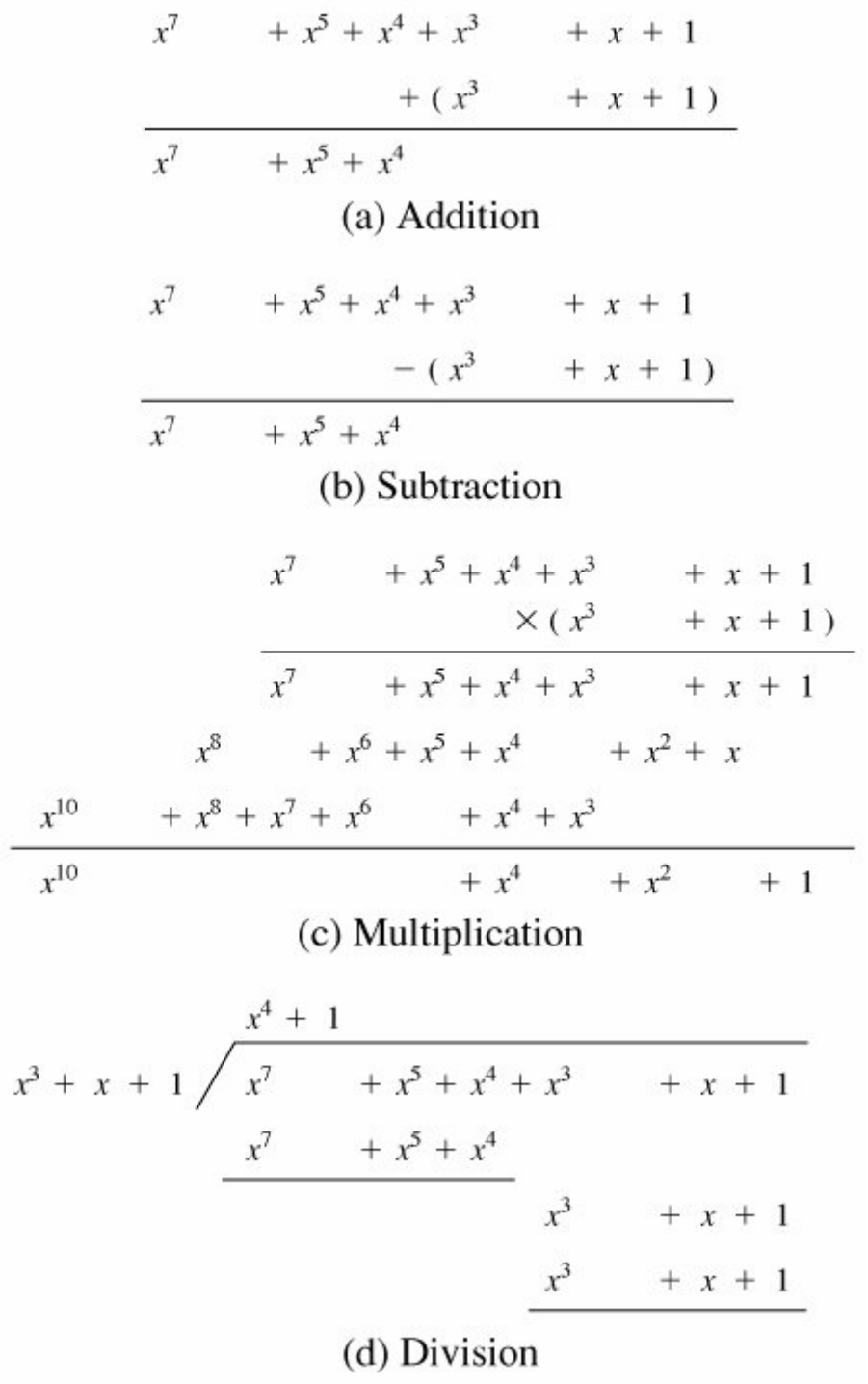


Figure 4.4. Examples of Polynomial Arithmetic over GF(2)

A polynomial f(x) over a field F is called irreducible if and only if f(x) cannot be expressed as a product of two polynomials, both over F, and both of degree lower than that of f(x). By analogy to integers, an irreducible polynomial is also called a **prime polynomial**.

The polynomial $f(x) = x^4 + 1$ over GF(2) is reducible, because $x^4 + 1 = (x + 1)(x^3 + x^2 + x + 1)$

[6] In the remainder of this chapter, unless otherwise noted, all examples are of polynomials over GF(2).

Consider the polynomial $f(x) = x^3 + x + 1$. It is clear by inspection that x is not a factor of f(x). We easily show that x + 1 is not a factor of f(x):

$$
\begin{array}{r}
x^2 + x \\
x + 1 \overline{)\, x^3 \qquad + x + 1} \\
\underline{x^3 + x^2} \\
\qquad x^2 + x \\
\qquad \underline{x^2 + x} \\
\qquad\qquad 1
\end{array}
$$

Thus f(x) has no factors of degree 1. But it is clear by inspection that if f(x) is reducible, it must have one factor of degree 2 and one factor of degree 1. Therefore, f(x) is irreducible.

4.5.3. Finding the Greatest Common Divisor

We can extend the analogy between polynomial arithmetic over a field and integer arithmetic by defining the greatest common divisor as follows. The polynomial c(x) is said to be the greatest common divisor of a(x) and b(x) if

1. c(x) divides both a(x) and b(x);
2. any divisor of a(x) and b(x) is a divisor of c(x).

An equivalent definition is the following: gcd[a(x), b(x)] is the polynomial of maximum degree that divides both a(x) and b(x).

We can adapt the Euclidean algorithm to compute the greatest common divisor of two polynomials. The equality in Equation (4.4) can be rewritten as the following theorem:

Equation 4-7

$$\gcd[a(x), b(x)] = \gcd[b(x), a(x) \bmod b(x)]$$

The Euclidean algorithm for polynomials can be stated as follows. The algorithm assumes that the degree of a(x) is greater than the degree of b(x). Then, to find gcd[a(x), b(x)],

EUCLID[a(x), b(x)]
1. A(x) ← a(x); B(x) ← b(x)
2. if B(x) = 0 return A(x) = gcd[a(x), b(x)]
3. R(x) = A(x) mod B(x)
4. A(x) ← B(x)
5. B(x) ← R(x)
6. goto 2

Find gcd[a(x), b(x)] for $a(x) = x^6 + x^5 + x^4 + x^3 + x^2 + x + 1$ and $b(x) = x^4 + x^2 + x + 1$.

A(x) = a(x); B(x) = b(x)

$$
\begin{array}{r}
x^2 + x \\
x^4 + x^2 + x + 1 \overline{)\, x^6 + x^5 + x^4 + x^3 + x^2 + x + 1} \\
\underline{x^6 \quad + x^4 + x^3 + x^2} \\
x^5 \quad + x + 1 \\
\underline{x^5 \quad + x^3 + x^2 + x} \\
x^3 + x^2 \quad + 1
\end{array}
$$

$R(x) = A(x) \bmod B(x) = x^3 + x^2 + 1$

$A(x) = x^4 + x^2 + x + 1$; $B(x) = x^3 + x^2 + 1$

$$
\begin{array}{r}
x + 1 \\
x^3 + x^2 + 1 \overline{)\, x^4 \quad + x^2 + x + 1} \\
\underline{x^4 + x^3 \quad + x} \\
x^3 + x^2 \quad + 1 \\
\underline{x^3 + x^2 \quad + 1}
\end{array}
$$

$R(x) = A(x) \bmod B(x) = 0$

$\gcd[a(x), b(x)] = A(x) = x^3 + x^2 + 1$

Summary

We began this section with a discussion of arithmetic with ordinary polynomials. In ordinary polynomial arithmetic, the variable is not evaluated; that is, we do not plug a value in for the variable of the polynomials. Instead, arithmetic operations are performed on polynomials (addition, subtraction, multiplication, division) using the ordinary rules of algebra. Polynomial division is not allowed unless the coefficients are elements of a field.

Next, we discussed polynomial arithmetic in which the coefficients are elements of GF(p). In this case, polynomial addition, subtraction, multiplication, and division are allowed. However, division is not exact; that is, in general division results in a quotient and a remainder.

Finally, we showed that the Euclidean algorithm can be extended to find the greatest common divisor of two polynomials whose coefficients are elements of a field.

All of the material in this section provides a foundation for the following section, in which polynomials are used to define finite fields of order p^n.

4.6. Finite Fields Of the Form GF(2^n)

Earlier in this chapter, we mentioned that the order of a finite field must be of the form p^n where p is a prime and n is a positive integer. In Section 4.4, we looked at the special case of finite fields with order p. We found that, using modular arithmetic in

Z_p, all of the axioms for a field (Figure 4.1) are satisfied. For polynomials over p^n, with $n > 1$, operations modulo p^n do not produce a field. In this section, we show what structure satisfies the axioms for a field in a set with p^n elements, and concentrate on GF(2^n).

4.6.1. Motivation

Virtually all encryption algorithms, both symmetric and public key, involve arithmetic operations on integers. If one of the operations that is used in the algorithm is division, then we need to work in arithmetic defined over a field. For convenience and for implementation efficiency, we would also like to work with integers that fit exactly into a given number of bits, with no wasted bit patterns. That is, we wish to work with integers in the range 0 through 2^n 1, which fit into an n-bit word.

Suppose we wish to define a conventional encryption algorithm that operates on data 8 bits at a time and we wish to perform division. With 8 bits, we can represent integers in the range 0 through 255. However, 256 is not a prime number, so that if arithmetic is performed in Z_{256} (arithmetic modulo 256), this set of integers will not be a field. The closest prime number less than 256 is 251. Thus, the set Z_{251}, using arithmetic modulo 251, is a field. However, in this case the 8-bit patterns representing the integers 251 through 255 would not be used, resulting in inefficient use of storage.

As the preceding example points out, if all arithmetic operations are to be used, and we wish to represent a full range of integers in n bits, then arithmetic modulo will not work; equivalently, the set of integers modulo 2^n, for $n > 1$, is not a field. Furthermore, even if the encryption algorithm uses only addition and multiplication, but not division, the use of the set Z_2^n is questionable, as the following example illustrates.

Suppose we wish to use 3-bit blocks in our encryption algorithm, and use only the operations of addition and multiplication. Then arithmetic modulo 8 is well defined, as shown in Table 4.1. However, note that in the multiplication table, the nonzero integers do not appear an equal number of times. For example, there are only four occurrences of 3, but twelve occurrences of 4. On the other hand, as was mentioned, there are finite fields of the form GF(2^n) so there is in particular a finite field of order $2^3 = 8$. Arithmetic for this field is shown in Table 4.5. In this case, the number of occurrences of the nonzero integers is uniform for multiplication. To summarize,

Integer	1	2	3	4	5	6	7
Occurrences in Z_8	4	8	4	12	4	8	4
Occurrences in GF(2^3)	7	7	7	7	7	7	7

Table 4.5. Arithmetic in GF(2^3)

+	000 (0)	001 (1)	010 (2)	011 (3)	100 (4)	101 (5)	110 (6)	111 (7)
000 (0)	0	1	2	3	4	5	6	7
001 (1)	1	0	3	2	5	4	7	6
010 (2)	2	3	0	1	6	7	4	5
011 (3)	3	2	1	0	7	6	5	4
100 (4)	4	5	6	7	0	1	2	3
101 (5)	5	4	7	6	1	0	3	2
110 (6)	6	7	4	5	2	3	0	1
111 (7)	7	6	5	4	3	2	1	0

(a) Addition

×	000 (0)	001 (1)	010 (2)	011 (3)	100 (4)	101 (5)	110 (6)	111 (7)
000 (0)	0	0	0	0	0	0	0	0
001 (1)	0	1	2	3	4	5	6	7
010 (2)	0	2	4	6	3	1	7	5
011 (3)	0	3	6	5	7	4	1	2
100 (4)	0	4	3	7	6	2	5	1
101 (5)	0	5	1	4	2	7	3	6
110 (6)	0	6	7	1	5	3	2	4
111 (7)	0	7	5	2	1	6	4	3

(b) Multiplication

w	$-w$	w^{-1}
0	0	—
1	1	1
2	2	5
3	3	6
4	4	7
5	5	2
6	6	3
7	7	4

(c) Additive and multiplicative inverses

For the moment, let us set aside the question of how the matrices of Table 4.5 were constructed and instead make some observations.

1. The addition and multiplication tables are symmetric about the main diagonal, in conformance to the commutative property of addition and multiplication. This property is also exhibited in Table 4.1, which uses mod 8 arithmetic.
2. All the nonzero elements defined by Table 4.5 have a multiplicative inverse, unlike the case with Table 4.1.
3. The scheme defined by Table 4.5 satisfies all the requirements for a finite field. Thus, we can refer to this scheme as GF(2^3).
4. For convenience, we show the 3-bit assignment used for each of the elements of GF(2^3).

Intuitively, it would seem that an algorithm that maps the integers unevenly onto themselves might be cryptographically weaker than one that provides a uniform mapping. Thus, the finite fields of the form GF(2^n) are attractive for cryptographic algorithms.

To summarize, we are looking for a set consisting of 2^n elements, together with a definition of addition and multiplication over the set that define a field. We can assign a unique integer in the range 0 through 2^n 1 to each element of the set. Keep in mind that we will not use modular arithmetic, as we have seen that this does not

result in a field. Instead, we will show how polynomial arithmetic provides a means for constructing the desired field.

4.6.2. Modular Polynomial Arithmetic

Consider the set S of all polynomials of degree n 1 or less over the field Z_p. Thus, each polynomial has the form

$$f(x) = a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \cdots + a_1x + a_0 = \sum_{i=0}^{n-1} a_i x^i$$

where each a_i takes on a value in the set {0, 1,..., p 1}. There are a total of p^n different polynomials in S.

For p = 3 and n = 2, the $3^2 = 9$ polynomials in the set are		
0	x	2x
1	x + 1	2x + 1
2	x + 2	2x + 2
For p = 2 and n = 3, the $2^3 = 8$ the polynomials in the set are		
0	x + 1	$x^2 + x$
1	x^2	$x^2 + x + 1$
x	$x^2 + 1$	

With the appropriate definition of arithmetic operations, each such set S is a finite field. The definition consists of the following elements:

1. Arithmetic follows the ordinary rules of polynomial arithmetic using the basic rules of algebra, with the following two refinements.
2. Arithmetic on the coefficients is performed modulo p. That is, we use the rules of arithmetic for the finite field Z_p.
3. If multiplication results in a polynomial of degree greater than n 1, then the polynomial is reduced modulo some irreducible polynomial m(x) of degree n. That is, we divide by m(x) and keep the remainder. For a polynomial f(x), the remainder is expressed as r(x) = f(x) mod m(x).

The Advanced Encryption Standard (AES) uses arithmetic in the finite field $GF(2^8)$, with the irreducible polynomial $m(x) = x^8 + x^4 x^3 + x + 1$. Consider the two polynomials $f(x) = x^6 + x^4 + x^2 + x + 1$ and $g(x) = x^7 + x + 1$. Then

$f(x) + g(x) = x^6 + x^4 x^2 + x + 1 + x^7 + x + 1$

$f(x) \times g(x) = x^{13} + x^{11} + x^9 + x^8 + x^7 +$
$x^7 + x^5 + x^3 + x^2 + x +$
$x^6 + x^4 + x^2 + x + 1$
$= x^{13} + x^{11} + x^9 + x^8 + x^6 + x^5 + x^4 + x^3 + 1$

$$
\begin{array}{r}
x^5 + x^3 \\
x^8 + x^4 + x^3 + x + 1 \overline{)\, x^{13} + x^{11} + x^9 + x^8 + x^7 + x^6 + x^5 + x^4 + x^3 + \quad + 1} \\
\underline{x^{13} \qquad + x^9 + x^8 \qquad + x^6 + x^5} \\
x^{11} \qquad\qquad + x^4 + x^3 \\
\underline{x^{11} \qquad + x^7 + x^6 \qquad + x^4 + x^3} \\
x^7 + x^6 \qquad\qquad + 1
\end{array}
$$

Therefore, $f(x) \times g(x) \bmod m(x) = x^7 + x^6 + 1$.

As with ordinary modular arithmetic, we have the notion of a set of residues in modular polynomial arithmetic. The set of residues modulo m(x), an nth-degree polynomial, consists of p^n elements. Each of these elements is represented by one of the p^n polynomials of degree $m < n$.

The residue class [x + 1], modulo m(x), consists of all polynomials a(x) such that $a(x) \in (x + 1) \pmod{m(x)}$. Equivalently, the residue class [x + 1] consists of all polynomials a(x) that satisfy the equality a(x) mod m(x)= x+1.

It can be shown that the set of all polynomials modulo an irreducible nth-degree polynomial m(x) satisfies the axioms in Figure 4.1, and thus forms a finite field. Furthermore, all finite fields of a given order are isomorphic; that is, any two finite-field structures of a given order have the same structure, but the representation, or labels, of the elements may be different.

To construct the finite field $GF(2^3)$, we need to choose an irreducible polynomial of degree 3. There are only two such polynomials: $(x^3 + x^2 + 1)$ and $(x^3 + x + 1)$. Using the latter, Table 4.6 shows the addition and multiplication tables for $GF(2^3)$. Note that this set of tables has the identical structure to those of Table 4.5. Thus, we have succeeded in finding a way to define a field of order 2^3.

Table 4.6. Polynomial Arithmetic Modulo $(x^3 + x + 1)$

+	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	0	1	x	x + 1	x^2	$x^2 + 1$	$x^2 + x$	$x^2 + x + 1$
001 1	1	0	x + 1	x	$x^2 + 1$	x^2	$x^2 + x + 1$	$x^2 + x$
010 x	x	x + 1	0	1	$x^2 + x$	$x^2 + x + 1$	x^2	$x^2 + 1$
011 x + 1	x + 1	x	1	0	$x^2 + x + 1$	$x^2 + x$	$x^2 + 1$	x^2
100 x^2	x^2	$x^2 + 1$	$x^2 + x$	$x^2 + x + 1$	0	1	x	x + 1
101 $x^2 + 1$	$x^2 + 1$	x^2	$x^2 + x + 1$	$x^2 + x$	1	0	x + 1	x
110 $x^2 + x$	$x^2 + x$	$x^2 + x + 1$	x^2	$x^2 + 1$	x	x + 1	0	1
111 $x^2 + x + 1$	$x^2 + x + 1$	$x^2 + x$	$x^2 + 1$	x^2	x + 1	x	1	0

(a) Addition

×	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	0	0	0	0	0	0	0	0
001 1	0	1	x	x + 1	x^2	$x^2 + 1$	$x^2 + x$	$x^2 + x + 1$
010 x	0	x	x^2	$x^2 + x$	x + 1	1	$x^2 + x + 1$	$x^2 + 1$
011 x + 1	0	x + 1	$x^2 + x$	$x^2 + 1$	$x^2 + x + 1$	x^2	1	x
100 x^2	0	x^2	x + 1	$x^2 + x + 1$	$x^2 + x$	x	$x^2 + 1$	1
101 $x^2 + 1$	0	$x^2 + 1$	1	x^2	x	$x^2 + x + 1$	x + 1	$x^2 + x$
110 $x^2 + x$	0	$x^2 + x$	$x^2 + x + 1$	1	$x^2 + 1$	x + 1	x	x^2
111 $x^2 + x + 1$	0	$x^2 + x + 1$	$x^2 + 1$	x	1	$x^2 + x$	x^2	x + 1

(b) Multiplication

4.6.3. Finding the Multiplicative Inverse

Just as the Euclidean algorithm can be adapted to find the greatest common divisor of two polynomials, the extended Euclidean algorithm can be adapted to find the multiplicative inverse of a polynomial. Specifically, the algorithm will find the multiplicative inverse of b(x) modulo m(x) if the degree of b(x) is less than the degree of m(x) and gcd[m(x), b(x)] = 1. If m(x) is an irreducible polynomial, then it has no factor other than itself or 1, so that gcd[m(x), b(x)] = 1. The algorithm is as follows:

EXTENDED EUCLID[m(x), b(x)]
1. [A1(x), A2(x), A3(x)] ← [1, 0, m(x)]; [B1(x), B2(x), B3(x)] ← [0, 1, b(x)]
2. if B3(x) = 0 return A3(x) = gcd[m(x), b(x)]; no inverse
3. if B3(x) = 1 return B3(x) = gcd[m(x), b(x)]; B2(x) = $b(x)^1$ mod m(x)
4. Q(x) = quotient of A3(x)/B3(x)
5. [T1(x), T2(x), T3(x)] ← [A1(x) Q(x)B1(x), A2(x) Q(x)B2(x), A3(x) QB3(x)]
6. [A1(x), A2(x), A3(x)] ← [B1(x), B2(x), B3(x)]
7. [B1(x), B2(x), B3(x)] ← [T1(x), T2(x), T3(x)]
8. goto 2

Table 4.7 shows the calculation of the multiplicative inverse of $(x^7 + x + 1)$ mod $(x^8 + x^4 + x^3 + x + 1)$. The result is that $(x^7 + x + 1)^1 = (x^7)$. That is, $(x^7 + x + 1)(x^7)$ 1 (mod $(x^8 + x^4 + x^3 + x + 1)$).

Table 4.7. Extended Euclid [$(x^8 + x^4 + x^3 + x + 1)$, $(x^7 + x + 1)$]

Initialization	A1(x) = 1; A2(x) = 0; A3(x) = $x^8 + x^4 + x^3 + x + 1$ B1(x) = 0; B2(x) = 1; B3(x) = $x^7 + x + 1$
Iteration 1	Q(x) = x A1(x) = 0; A2(x) = 1; A3(x) = $x^7 + x + 1$ B1(x) = 1; B2(x) = x; B3(x) = $x^4 + x^3 + x^2 + 1$
Iteration 2	Q(x) = $x^3 + x^2 + 1$ A1(x) = 1; A2(x) = x; A3(x) = $x^4 + x^3 + x^2 + 1$ B1(x) = $x^3 + x^2 + 1$; B2(x) = $x^4 + x^3 + x + 1$; B3(x) = x
Iteration 3	Q(x) = $x^3 + x^2 + x$ A1(x) = $x^3 + x^2 + 1$; A2(x) = $x^4 + x^3 + x + 1$; A3(x) = x B1(x) = $x^6 + x^2 + x + 1$; B2(x) = x^7; B3(x) = 1
Iteration 4	B3(x) = gcd[$(x^7 + x + 1)$, $(x^8 + x^4 + x^3 + x + 1)$] = 1 B2(x) = $(x^7 + x + 1)^1$ mod $(x^8 + x^4 + x^3 + x + 1) = x^7$

4.6.4. Computational Considerations

A polynomial f(x) in GF(2^n)

$$f(x) = a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \cdots + a_1x + a_0 = \sum_{i=0}^{n-1} a_i x^i$$

can be uniquely represented by its n binary coefficients ($a_{n1}a_{n2}...a_0$). Thus, every polynomial in GF(2^n) can be represented by an n-bit number.

Tables 4.5 and 4.6 show the addition and multiplication tables for GF(2^3) modulo m(x) = (x^3 + x + 1). Table 4.5 uses the binary representation, and Table 4.6 uses the polynomial representation.

Addition

We have seen that addition of polynomials is performed by adding corresponding coefficients, and, in the case of polynomials over Z_2 addition is just the XOR operation. So, addition of two polynomials in GF(2^n) corresponds to a bitwise XOR operation.

Consider the two polynomials in GF(2^8) from our earlier example: f(x) = x^6 + x^4 + x^2 + x + 1 and g(x) = x^7 + x + 1.

(x^6 + x^4 + x^2 + x + 1) + (x^7+ x + 1)	= x^7 + x^6 + x^6 + x^4 + x^2	(polynomial notation)
(01010111) ⊕ (10000011)	= (11010100)	(binary notation)
{57} ⊕ {83}	= {D4}	(hexadecimal notation)

Multiplication

There is no simple XOR operation that will accomplish multiplication in GF(2^n) However, a reasonably straightforward, easily implemented technique is available. We will discuss the technique with reference to GF(2^8) using m(x) = x^8 + x^4 + x^3 + x + 1, which is the finite field used in AES. The technique readily generalizes to GF(2^n).

The technique is based on the observation that

Equation 4-8

$$x^8 \bmod m(x) = [m(x) - x^8] = (x^4 + x^3 + x + 1)$$

A moment's thought should convince you that Equation (4.8) is true; if not, divide it out. In general, in GF(2^n) with an nth-degree polynomial p(x), we have x^n mod p(x) = [p(x) x^n].

Now, consider a polynomial in GF(2^8), which has the form f(x) = b_7x^7 + b_6x^6 + b_5x^5 + b_4x^4 + b_3x^3 + b_2x^2 + b_1x + b_0. If we multiply by x, we have

Equation 4-9

$$x \times f(x) = (b_7x^8 + b_6x^7 + b_5x^6 + b_4x^5 + b_3x^4 + b_2x^3 + b_1x^2 + b_0x) \bmod m(x)$$

If $b_7 = 0$, then the result is a polynomial of degree less than 8, which is already in reduced form, and no further computation is necessary. If $b_7 = 1$, then reduction modulo m(x) is achieved using Equation (4.8):
x x f(x) = $(b_6x^7 + b_5x^6 + b_4x^5 + b_3x^4 + b_2x^3 +$
$b_1x^2 + b_0x) + (x^4 + x^3 + x + 1)$
It follows that multiplication by x (i.e., 00000010) can be implemented as a 1-bit left shift followed by a conditional bitwise XOR with (00011011), which represents ($x^4 +$ $x^3 + x + 1$). To summarize,
Equation 4-10

$$x \times f(x) = \begin{cases} (b_6b_5b_4b_3b_2b_1b_00) & \text{if } b_7 = 0 \\ (b_6b_5b_4b_3b_2b_1b_00) \oplus (00011011) & \text{if } b_7 = 1 \end{cases}$$

Multiplication by a higher power of x can be achieved by repeated application of Equation (4.10). By adding intermediate results, multiplication by any constant in GF(2^8) can be achieved.

In an earlier example, we showed that for $f(x) = x^6 + x^4 + x^2 + x + 1$, $g(x) = x^7 + x + 1$, and $m(x) = x^8 + x^4 + x^3 + x + 1$, f(x) x g(x) mod m(x) = $x^7 + x^6 +$ 1. Redoing this in binary arithmetic, we need to compute (01010111) × (10000011). First, we determine the results of multiplication by powers of x:
(01010111) × (00000001) = (10101110)
(01010111) × (00000100) = (01011100) ⊕ (00011011) = (01000111)
(01010111) × (00001000) = (10001110)
(01010111) × (00010000) = (00011100) ⊕ (00011011) = (00000111)
(01010111) × (00100000) = (00001110)
(01010111) × (01000000) = (00011100)
(01010111) × (10000000) = (00111000)
So,
(01010111) × (10000011) = (01010111) × [(00000001) × (00000010) × (10000000)]
= (01010111) ⊕ (10101110) ⊕ (00111000) = (11000001)
which is equivalent to $x^7 + x^6 + 1$.

Using a Generator

An equivalent technique for defining a finite field of the form GF(2^n) using the same irreducible polynomial, is sometimes more convenient. To begin, we need two definitions: A generator g of a finite field F of order q (contains q elements) is an element whose first q 1 powers generate all the nonzero elements of F. That is, the elements of F consist of 0, g^0, g^1,..., g^{q2}. Consider a field F defined by a polynomial f(x). An element b contained in F is called a root of the polynomial if f(b) = 0. Finally, it can be shown that a root g of an irreducible polynomial is a generator of the finite field defined on that polynomial.

Let us consider the finite field GF(2^3), defined over the irreducible

polynomial $x^3 + x + 1$, discussed previously. Thus, the generator g must satisfy $f(x) = g^3 + g + 1 = 0$. Keep in mind, as discussed previously, that we need not find a numerical solution to this equality. Rather, we deal with polynomial arithmetic in which arithmetic on the coefficients is performed modulo 2. Therefore, the solution to the preceding equality is $g^3 = g\ 1 = g + 1$. We now show that g in fact generates all of the polynomials of degree less than 3. We have the following:

$g^4 = g(g^3) = g(g + 1) = g^2 + g$
$g^5 = g(g^4) = g(g^2 + g) = g^3 + g^2 = g^2 + g + 1$
$g^6 = g(g^5) = g(g^2 + g + 1) = g^3 + g^2 + g = g^2 + g + g + 1 = g^2 + 1$
$g^7 = g(g^6) = g(g^2 + 1) = g^3 + g = g + g + 1 = 1 = g^0$

We see that the powers of g generate all the nonzero polynomials in $GF(2^3)$. Also, it should be clear that $g^k = g^{k \bmod 7}$ for any integer k. Table 4.8 shows the power representation, as well as the polynomial and binary representations.

Table 4.8. Generator for $GF(2^3)$ using $x^3 + x + 1$

Power Representation	Polynomial Representation	Binary Representation	Decimal (Hex) Representation
0	0	000	0
g^0 (= g^7)	1	001	1
g^1	g	010	2
g^2	g^2	100	4
g^3	g + 1	011	3
g^4	g^2 + g	110	6
g^5	g^2 + g + 1	111	7
g^6	g^2 + 1	101	5

This power representation makes multiplication easy. To multiply in the power notation, add exponents modulo 7. For example, $g^4 \times g^6 = g^{(10 \bmod 7)} = g^3 = g + 1$. The same result is achieved using polynomial arithmetic, as follows: we have $g^4 = g^2 + g$ and $g^6 = g^2 + 1$. Then, $(g^2 + g) \times (g^2 + 1) = g^4 + g^3 + g^2 + 1$. Next, we need to determine $(g^4 + g^3 + g^2 + 1) \bmod (g^3 + g + 1)$ by division:

$$
\begin{array}{r}
g + 1 \\
g^3 + g^2 + 1 \overline{)\, g^4 + g^3 + g^2 + g} \\
\underline{g^4 + \quad g^2 + g} \\
g^3 \\
\underline{g^3 + \quad g + 1} \\
g + 1
\end{array}
$$

We get a result of g + 1, which agrees with the result obtained using the power representation.

Table 4.9 shows the addition and multiplication tables for GF(2^3) using the power represenation. Note that this yields the identical results to the polynomial representation (Table 4.6) with some of the rows and columns interchanged.

Table 4.9. GF(2^3) Arithmetic Using Generator for the Polynomial ($x^3 + x + 1$)

+	000 0	001 1	010 g	100 g^2	011 g^3	110 g^4	111 g^5	101 g^6
000 0	0	1	g	g^2	$g+1$	g^2+g	g^2+g+1	g^2+1
001 1	1	0	$g+1$	g^2+1	g	g^2+g+1	g^2+g	g^2
010 g	g	$g+1$	0	g^2+g	1	g^2	g^2+1	g^2+g+1
100 g^2	g^2	g^2+1	g^2+g	0	g^2+g+1	g	$g+1$	1
011 g^3	$g+1$	g	1	g^2+g+1	0	g^2+1	g^2	g^2+g
110 g^4	g^2+g	g^2+g+1	g^2	g	g^2+1	0	1	$g+1$
111 g^5	g^2+g+1	g^2+g	g^2+1	$g+1$	g^2	1	0	g
101 g^6	g^2+1	g^2	g^2+g+1	1	g^2+g	$g+1$	g	0

(a) Addition

×	000 0	001 1	010 g	100 g^2	011 g^3	110 g^4	111 g^5	101 g^6
000 0	0	0	0	0	0	0	0	0
001 1	0	1	g	g^2	$g+1$	g^2+g	g^2+g+1	g^2+1
010 g	0	g	g^2	$g+1$	g^2+g	g^2+g+1	g^2+1	1
100 g^2	0	g^2	$g+1$	g^2+g	g^2+g+1	g^2+1	1	g
011 g^3	0	$g+1$	g^2+g	g^2+g+1	g^2+1	1	G	g^2
110 g^4	0	g^2+g	g^2+g+1	g^2+1	1	g	g^2	$g+1$
111 g^5	0	g^2+g+1	g^2+1	1	g	g^2	$g+1$	g^2+g
101 g^6	0	g^2+1	1	g	g^2	$g+1$	g^2+g	g^2+g+1

(b) Multiplication

In general, for GF(2^n) with irreducible polynomial f(x), determine $g^n = f(x)$ g^n. Then calculate all of the powers of g from g^{n+1} through g^{2n2}. The elements of the field correspond to the powers of g from through g^{2n2}, plus the value 0. For multiplication of two elements in the field, use the equality $g^k = g^{k \bmod (2n1)}$ for any integer k.

Summary

In this section, we have shown how to construct a finite field of order 2^n. Specifically, we defined GF(2^n) with the following properties:

1. GF(2^n) consists of 2^n elements.
2. The binary operations + and x are defined over the set. The operations of addition, subtraction, multiplication, and division can be performed without leaving the set. Each element of the set other than 0 has a multiplicative inverse.

We have shown that the elements of GF(2^n) can be defined as the set of all polynomials of degree n 1 or less with binary coefficients. Each such polynomial can be represented by a unique n-bit value. Arithmetic is defined as polynomial arithmetic modulo some irreducible polynomial of degree n. We have also seen that an equivalent definition of a finite field GF(2^n) makes use of a generator and that arithmetic is defined using powers of the generator.