$pq) = (1/q) + (1/p) - (1/(pq))$, a very small number if p and q

digits.]

er $p = 61$, $q = 127$, with $n = (61)(127) = 7747$ and $r = \phi(61)$ ·

7560. Now suppose we select e as 17.

ng message that we wish to encrypt.

INVEST IN BONDS

assignments as in part (b) of Example 14.15, here we would replace

merely 8). Then we replace "N" by 13. This provides us with the

— namely, 0813 — for the first two letters "IN". The assignment

e is as follows [where we have appended the letter "X" to the right

l block to have two letters (or, four digits)]:

E	S	T	I	N	B	O	N	D	S	X
04	18	19	08	13	01	14	13	03	18	23

ock B of four digits by the encryption function E, where $E(B) =$

ar exponentiation can be carried out efficiently by using the proce-

) So here the domain of E is the concatenation of $\mathbf{Z}_{26}$ with itself,

69 2104^{17} **mod** 7747 = 0628 1819^{17} **mod** 7747 = 5540

69 0114^{17} **mod** 7747 = 6560 1303^{17} **mod** 7747 = 6401

329.

ent of the encrypted assignment (for the given plaintext message)

0628 5540 2169 6560 6401 4829.

ow does the recipient decrypt the ciphertext received?"

$_r$ $(= \mathbf{Z}_{\phi(n)})$, we can use the Euclidean algorithm (as in Example

$= d$. Then we define the decryption function D, where $D(C) =$

C of four digits. Since $e^{-1} = d$, it follows that $ed \equiv 1 \bmod \phi(n)$ —

1. Therefore, $ed = k\phi(n) + 1$, for some $k \in \mathbf{Z}$. Now recall the ar-

or the probability that a randomly selected element e from $\mathbf{Z}_n$ is

$_n$). For any block B of four digits, we consider B as an element of

er B as a unit in $\mathbf{Z}_n$. Since the units in the ring $(\mathbf{Z}_n, +, \cdot)$ form a

er multiplication, it follows from the result in Exercise 8 of Section

$^{-1} = (B^{\phi(n)})^k B^1 \equiv B \pmod{n}$, or B^{ed} **mod** $n = B$. [This is also a

Theorem, as stated in part (b) of Exercise 13 in Section 16.3.]

rom the previous paragraph in our example we have $p = 61$, $q =$

$= \phi(n) = (p-1)(q-1) = (60)(126) = 7560$, and $e = 17$. From

n we calculate $d = e^{-1} = 3113$. Now we find, for instance, that

813 and that 0628^{3113} **mod** $7747 = 2104$. Continuing, the recipient

assignment for the original plaintext and then the plaintext.

RSA cryptosystem more secure than the private-key cryptosystems

hould relate that the RSA cryptosystem is *not* a private-key cryp-

s an example of a *public-key* cryptosystem, where the key (n, e) is

ns that all one needs to do to decrypt the encrypted assignment is

to determine $d = e^{-1}$ in $\mathbf{Z}_r$ $(= \mathbf{Z}_{\phi(n)})$. Now it is time to realize that by knowing n we do not immediately know r. For to be able to determine $r = (p-1)(q-1)$, we need to know p, q, the prime factors of n. And this is what makes this system so much more secure than the other cryptosystems we mentioned. Determining the primes p, q, when they are 100 or more digits long, is not a feasible problem. However, as computer power continues to improve, to keep the RSA cryptosystem secure, one may need to redefine the key using primes with more and more digits.

In closing, we show how the problem of factoring the modulus n as pq is related to the problem of determining $r = (p-1)(q-1)$. We start by observing that

$$p + q = pq - (p-1)(q-1) + 1 = n - \phi(n) + 1 = n - r + 1,$$

while

$$p - q = \sqrt{(p-q)^2} = \sqrt{(p-q)^2 + 4pq - 4pq} = \sqrt{(p+q)^2 - 4pq}$$
$$= \sqrt{(p+q)^2 - 4n} = \sqrt{(n-r+1)^2 - 4n}.$$

Then, from these two equations, we learn that

$$p = (1/2)[(p+q) + (p-q)] = (1/2)[(n-r+1) + \sqrt{(n-r+1)^2 - 4n}]$$

and

$$q = (1/2)[(p+q) - (p-q)] = (1/2)[(n-r+1) - \sqrt{(n-r+1)^2 - 4n}].$$

Consequently, when we know n and r, then we can readily determine the primes p, q such that $n = pq$.

EXERCISES 16.4

The use of a computer algebra system is strongly recommended for the first four exercises.

1. Determine the ciphertext for the plaintext INVEST IN STOCKS, when using RSA encryption with $e = 7$ and $n = 2573$.

2. Determine the ciphertext for the plaintext ORDER A PIZZA, when using RSA encryption with $e = 5$ and $n = 1459$.

3. Determine the plaintext for the RSA ciphertext 1418 1436 2370 1102 1805 0250, if $e = 11$ and $n = 2501$.

4. Determine the plaintext for the RSA ciphertext 0986 3029 1134 1105 1232 2281 2967 0272 1818 2398 1153, if $e = 17$ and $n = 3053$.

5. Find the primes p, q if $n = pq = 121{,}361$ and $\phi(n) = 120{,}432$.

6. Find the primes p, q if $n = pq = 5{,}446{,}367$ and $\phi(n) = 5{,}441{,}640$.

16.5 Elements of Coding Theory

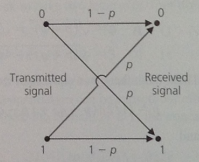
In this and the next four sections we introduce an area of applied mathematics called *algebraic coding theory*. This theory was inspired by the fundamental paper of Claude Shannon (1948) along with results by Marcel Golay (1949) and Richard Hamming (1950). Since that time it has become an area of great interest where algebraic structures, probability, and combinatorics all play a role.

Our coverage will be held to an introductory level as we seek to model the transmission of information represented by strings of the signals 0 and 1.

In digital communications, when information is transmitted in the form of strings of 0's and 1's, certain problems arise. As a result of "noise" in the channel, when a certain signal is transmitted a different signal may be received, thus causing the receiver to make a wrong

decision. Hence we want to develop techniques to help us detect, and perhaps even correct, transmission errors. However, we can only improve the chances of correct transmission; there are no guarantees.

Our model uses a *binary symmetric channel*, as shown in Fig. 16.2. The adjective *binary* appears because an individual signal is represented by one of the bits 0 or 1. When a transmitter sends the signal 0 or 1 in such a channel, associated with either signal is a (constant) probability p for incorrect transmission. When that probability p is the same for both signals, the channel is called *symmetric*. Here, for example, we have probability p of sending 0 and having 1 received. The probability of sending signal 0 and having it received correctly is then $1 - p$. All possibilities are illustrated in Fig. 16.2.



The Binary Symmetric Channel

Figure 16.2

6.19

Consider the string $c = 10110$. We regard c as an element of the group $\mathbf{Z}_2^5$, formed from the direct product of five copies of $(\mathbf{Z}_2, +)$. To shorten notation we write 10110 instead of $(1, 0, 1, 1, 0)$. When sending each bit (individual signal) of c through the binary symmetric channel, we assume that the probability of incorrect transmission is $p = 0.05$, so that the probability of transmitting c with no errors is $(0.95)^5 \doteq 0.77$.

Here, and throughout our discussion of coding theory, we assume that the transmission of each signal does not depend in any way on the transmissions of prior signals. Consequently, the probability of the occurrence of all of these *independent* events (in their prescribed order) is given by the product of their individual probabilities.

What is the probability that the party receiving the five-bit message receives the string $r = 00110$ — that is, the original message with an error in the first position? The probability of incorrect transmission for the first bit is 0.05, so with the assumption of independent events, $(0.05)(0.95)^4 \doteq 0.041$ is the probability of sending $c = 10110$ and receiving $r = 00110$. With $e = 10000$, we can write $c + e = r$ and interpret r as the result of the sum of the original message c and the particular *error pattern* $e = 10000$. Since $c, r, e \in \mathbf{Z}_2^5$ and $-1 = 1$ in $\mathbf{Z}_2$, we also have $c + r = e$ and $r + e = c$.

In transmitting $c = 10110$, the probability of receiving $r = 00100$ is

$$(0.05)(0.95)^2(0.05)(0.95) \doteq 0.002,$$

so this multiple error is not very likely to occur.

Finally if we transmit $c = 10110$, what is the probability that r differs from c in exactly two places? To answer this we sum the probabilities for each error pattern consisting of two 1's and three 0's. Each such pattern has probability 0.002. There are $\binom{5}{2}$ such patterns, so

the probability of two errors in transmission is given by

$$\binom{5}{2}(0.05)^2(0.95)^3 \doteq 0.021.$$

These results lead us to the following theorem.

THEOREM 16.10

Let $c \in \mathbf{Z}_2^n$. For the transmission of c through a binary symmetric channel with probability p of incorrect transmission,

a) the probability of receiving $r = c + e$, where e is a *particular* error pattern consisting of k 1's and $(n - k)$ 0's, is $p^k(1 - p)^{n-k}$.

b) the probability that (exactly) k errors are made in the transmission is

$$\binom{n}{k} p^k (1 - p)^{n-k}.^{\dagger}$$

In Example 16.19, the probability of making at most one error in the transmission of $c = 10110$ is $(0.95)^5 + \binom{5}{1}(0.05)(0.95)^4 \doteq 0.977$. Thus the chance for multiple errors in transmission will be considered negligible throughout the discussion in this chapter. Such an assumption is valid when p is small. In actuality, a binary symmetric channel is considered "good" when $p < 10^{-5}$. However, no matter what else we stipulate, we always want $p < 1/2$.

To improve the accuracy of transmission in a binary symmetric channel, certain types of coding schemes can be used where extra bits are provided.

For $m, n \in \mathbf{Z}^+$, let $n > m$. Consider $\emptyset \neq W \subseteq \mathbf{Z}_2^m$. The set W consists of the *messages* to be transmitted. To each $w \in W$ are appended $n - m$ extra bits to form the *code word* c, where $c \in \mathbf{Z}_2^n$. This process is called *encoding* and is represented by the function $E: W \to \mathbf{Z}_2^n$. Then $E(w) = c$ and $E(W) = C \subseteq \mathbf{Z}_2^n$. Since the function E simply appends extra bits to the (distinct) messages, the encoding process is one-to-one. Upon transmission, c is received as $T(c)$, where $T(c) \in \mathbf{Z}_2^n$. Unfortunately, T is not a function because $T(c)$ may be different at different transmission times (for the noise in the channel changes with time). (See Fig. 16.3.)

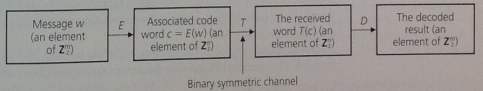


Figure 16.3

Upon receiving $T(c)$, we want to apply a decoding function $D: \mathbf{Z}_2^n \to \mathbf{Z}_2^m$ to remove the extra bits and, we hope, obtain the original message w. Ideally $D \circ T \circ E$ should be the identity function on W, with $D: C \to W$. Since this cannot be expected, we seek functions E and D such that there is a high probability of correctly decoding the received word $T(c)$ and recapturing the original message w. In addition, we want the ratio m/n to be as large as possible so that an excessive number of bits are not appended to w in getting the code

†This is the binomial probability distribution that was developed in (optional) Sections 3.5 and 3.7.

word $c = E(w)$. This ratio m/n measures the *efficiency* of our scheme and is called the *rate* of the code. Finally, the functions E and D should be more than theoretical results; they must be practical in the sense that they can be implemented electronically.

In such a scheme, the functions E and D are called the *encoding* and *decoding* functions, respectively, of an (n, m) *block code*.

We illustrate these ideas in the following two examples.

EXAMPLE 16.20

Consider the $(m + 1, m)$ block code for $m = 8$. Let $W = \mathbf{Z}_2^8$. For each $w = w_1w_2 \cdots w_8 \in W$, define $E\colon \mathbf{Z}_2^8 \to \mathbf{Z}_2^9$ by $E(w) = w_1w_2 \cdots w_8w_9$, where $w_9 = \sum_{i=1}^{8} w_i$, with the addition performed modulo 2. For example, $E(11001101) = 110011011$, and $E(00110011) = 001100110$.

For all $w \in \mathbf{Z}_2^8$, $E(w)$ contains an even number of 1's. So for $w = 11010110$ and $E(w) = 110101101$, if we receive $T(c) = T(E(w))$ as 100101101, from the odd number of 1's in $T(c)$ we know that a mistake has occurred in transmission. Hence we are able to *detect* single errors in transmission. But we seem to have no way to correct such errors.

The probability of sending the code word 110101101 and making at most one error in transmission is

$$\underbrace{(1-p)^9}_{\substack{\text{All nine bits are}\\ \text{correctly transmitted.}}} + \underbrace{\binom{9}{1}p(1-p)^8}_{\substack{\text{One bit is changed in}\\ \text{transmission and an error is detected.}}}.$$

For $p = 0.001$ this gives $(0.999)^9 + \binom{9}{1}(0.001)(0.999)^8 \doteq 0.99996417$.

If we detect an error and we are able to relay a signal back to the transmitter to repeat the transmission of the code word, and continue this process until the received word has an even number of 1's, then the probability of sending the code word 110101101 and receiving the correct transmission is approximately 0.99996393.†

Should an even positive number of errors occur in transmission, $T(c)$ is unfortunately accepted as the correct code word and we interpret its first eight components as the original message. This scheme is called the $(m + 1, m)$ *parity-check code* and is appropriate only when multiple errors are not likely to occur.

If we send the message 11010110 through the channel, we have probability $(0.999)^8 = 0.99202794$ of correct transmission. By using this parity-check code, we increase our chances of getting the correct message to (approximately) 0.99996393. However, an extra signal is sent (and perhaps additional transmissions are needed) and the rate of the code has decreased from 1 to 8/9.

But suppose that instead of sending eight bits we sent 160 bits, in successive strings of length 8. The chances of receiving the correct message without any coding scheme would be $(0.999)^{160} \doteq 0.85207557$. With the parity-check method we send 180 bits, but the chances for correct transmission now increase to $(0.999964)^{20} \doteq 0.99928025$.

†For $p = 0.001$ the probability that an odd number of errors occurs in the transmission of the code word 110101101 is

$$p_{\text{odd}} = \binom{9}{1}(0.999)^8(0.001) + \binom{9}{3}(0.999)^6(0.001)^3 + \binom{9}{5}(0.999)^4(0.001)^5 + \binom{9}{7}(0.999)^2(0.001)^7 + \binom{9}{9}(0.001)^9$$
$$\doteq 0.008928251 + 0.000000083 + 0.000000000 + 0.000000000 + 0.000000000 = 0.008928334.$$

With q = the probability of the correct transmission of 110101101 = $(0.999)^9$, the probability that this code word is transmitted and correctly received under these conditions (of retransmission) is then given by

$$q + p_{\text{odd}} \cdot q + (p_{\text{odd}})^2 q + (p_{\text{odd}})^3 q + \cdots = q/(1 - p_{\text{odd}}) \doteq 0.99996393 \text{ (to eight decimal places)}.$$

EXAMPLE 16.21

The $(3m, m)$ *triple repetition code* is one where we can both *detect* and *correct* single errors in transmission. With $m = 8$ and $W = \mathbf{Z}_2^8$, we define $E\colon \mathbf{Z}_2^8 \to \mathbf{Z}_2^{24}$ by $E(w_1w_2 \cdots w_8) = w_1w_2 \cdots w_8w_1w_2 \cdots w_8w_1w_2 \cdots w_8$.

Hence if $w = 10110111$, then $c = E(w) = 101101111101101111010110111$.

The decoding function $D\colon \mathbf{Z}_2^{24} \to \mathbf{Z}_2^8$ is carried out by the majority rule. For example, if $T(c) = 101001110011011110110110$, then we have three errors occurring in positions 4, 9, and 24. We decode $T(c)$, by examining the first, ninth, and seventeenth positions to see which signal appears more times. Here it is 1 (which occurs twice), so we decode the first entry in the decoded message as 1. Continuing with the entries in the second, tenth, and eighteenth positions, the result for the second entry of the decoded message is 0 (which occurs all three times). As we proceed, we recapture the correct message, 10110111.

Although we have more than one transmission error here, all is well unless two (or more) errors occur with the second error eight or sixteen spaces after the first — that is, if two (or more) incorrect transmissions occur for the same bit of the original message.

Now how does this scheme compare with the other methods we have? With $p = 0.001$, the probability of correctly decoding a single bit is $(0.999)^3 + \binom{3}{1}(0.001)(0.999)^2 \doteq 0.99999700$. So the probability of receiving and correctly decoding the eight-bit message is $(0.99999700)^8 = 0.99997600$, just slightly better than the result from the parity-check method (where we may have to retransmit, thus increasing the overall transmission time). Here we transmit 24 signals for this message, so our rate is now 1/3. For this increased accuracy and the ability to detect and now *correct* single errors (which we could not do in any previous schemes), we may pay with an increase in transmission time. But we do not waste time with retransmissions.

EXERCISES 16.5

1. Let C be a set of code words, where $C \subseteq \mathbf{Z}_2^7$. In each of the following, two of e (error pattern), r (received word) and c (code word) are given, with $r = c + e$. Determine the third term.

a) $c = 1010110$, $r = 1011111$

b) $c = 1010110$, $e = 0101101$

c) $e = 0101111$, $r = 0000111$

2. A binary symmetric channel has probability $p = 0.05$ of incorrect transmission. If the code word $c = 011011101$ is transmitted, what is the probability that (a) we receive $r = 011111101$? (b) we receive $r = 111011100$? (c) a single error occurs? (d) a double error occurs? (e) a triple error occurs? (f) three errors occur, no two of them consecutive?

3. Let $E\colon \mathbf{Z}_2^3 \to \mathbf{Z}_2^5$ be the encoding function for the (9, 3) triple repetition code.

a) If $D\colon \mathbf{Z}_2^5 \to \mathbf{Z}_2^3$ is the corresponding decoding function, apply D to decode the received words (i) 111101100; (ii) 000100011; (iii) 010011111.

b) Find three different received words r for which $D(r) = 000$.

c) For each $w \in \mathbf{Z}_2^3$, what is $|D^{-1}(w)|$?

4. The $(5m, m)$ five-times repetition code has encoding function $E\colon \mathbf{Z}_2^m \to \mathbf{Z}_2^{5m}$, where $E(w) = wwwww$. Decoding with $D\colon \mathbf{Z}_2^{5m} \to \mathbf{Z}_2^m$ is accomplished by the majority rule. (Here we are able to correct single and double errors made in transmission.)

a) With $p = 0.05$, what is the probability for the transmission and correct decoding of the signal 0?

b) Answer part (a) for the message 110 in place of the signal 0.

c) For $m = 2$, decode the received word

$$r = 0111001001.$$

d) If $m = 2$, find three received words r where $D(r) = 00$.

e) For $m = 2$ and $D\colon \mathbf{Z}_2^{10} \to \mathbf{Z}_2^2$, what is $|D^{-1}(w)|$ for each $w \in \mathbf{Z}_2^2$?

16.6 The Hamming Metric

In this section we develop the general principles for discussing the error-detecting and error-correcting capabilities of a coding scheme. These ideas were developed by Richard Wesley Hamming (1915–1998).

We start by considering a code $C \subseteq \mathbf{Z}_2^4$, where $c_1 = 0111$, $c_2 = 1111 \in C$. Now both the transmitter and the receiver know the elements of C. So if the transmitter sends c_1 but the person receiving the code word receives $T(c_1)$ as 1111, then he or she feels that c_2 was transmitted and makes whatever decision (a wrong one) c_2 implies. Consequently, although only one transmission error was made, the results could be unpleasant. Why is this? Unfortunately we have two code words that are almost the same. They are rather *close* to each other, for they differ in only one component.

We describe this notion of closeness more precisely as follows.

Definition 16.9 For each element $x = x_1x_2 \cdots x_n \in \mathbf{Z}_2^n$, where $n \in \mathbf{Z}^+$, the *weight of* x, denoted $\text{wt}(x)$, is the number of components x_i of x, for $1 \le i \le n$, where $x_i = 1$. If $y \in \mathbf{Z}_2^n$, the *distance between* x *and* y, denoted $d(x, y)$, is the number of components where $x_i \ne y_i$, for $1 \le i \le n$.

EXAMPLE 16.22 For $n = 5$, let $x = 01001$ and $y = 11101$. Then $\text{wt}(x) = 2$, $\text{wt}(y) = 4$, and $d(x, y) = 2$. In addition, $x + y = 10100$, so $\text{wt}(x + y) = 2$. Is it just by chance that $d(x, y) = \text{wt}(x + y)$? For each $1 \le i \le 5$, $x_i + y_i$ contributes a count of 1 to $\text{wt}(x + y) \iff x_i \ne y_i \iff x_i, y_i$ contribute a count of 1 to $d(x, y)$. [This is actually true for all $n \in \mathbf{Z}^+$, so $\text{wt}(x + y) = d(x, y)$ for all $x, y \in \mathbf{Z}_2^n$.]

When $x, y \in \mathbf{Z}_2^n$, we write $d(x, y) = \sum_{i=1}^{n} d(x_i, y_i)$ where,

$$\text{for each } 1 \le i \le n, \qquad d(x_i, y_i) = \begin{cases} 0 & \text{if } x_i = y_i \\ 1 & \text{if } x_i \ne y_i. \end{cases}$$

LEMMA 16.2 For all $x, y \in \mathbf{Z}_2^n$, $\text{wt}(x + y) \le \text{wt}(x) + \text{wt}(y)$.

Proof: We prove this lemma by examining, for each $1 \le i \le n$, the components $x_i, y_i, x_i + y_i$, of $x, y, x + y$, respectively. Only one situation would cause this inequality to be false: if $x_i + y_i = 1$ while $x_i = 0$ and $y_i = 0$, for some $1 \le i \le n$. But this never occurs because $x_i + y_i = 1$ implies that exactly one of x_i and y_i is 1.

In Example 16.22 we found that

$$\text{wt}(x + y) = \text{wt}(10100) = 2 \le 2 + 4 = \text{wt}(01001) + \text{wt}(11101) = \text{wt}(x) + \text{wt}(y).$$

THEOREM 16.11 The distance function d defined on $\mathbf{Z}_2^n \times \mathbf{Z}_2^n$ satisfies the following for all $x, y, z \in \mathbf{Z}_2^n$.

a) $d(x, y) \ge 0$

b) $d(x, y) = 0 \iff x = y$

c) $d(x, y) = d(y, x)$

d) $d(x, z) \le d(x, y) + d(y, z)$

Proof: We leave the first three parts for the reader and prove part (d).

In $\mathbf{Z}_2^n$, $y + y = 0$, so $d(x, z) = \text{wt}(x + z) = \text{wt}(x + (y + y) + z) = \text{wt}((x + y) + (y + z)) \le \text{wt}(x + y) + \text{wt}(y + z)$, by Lemma 16.2. With $\text{wt}(x + y) = d(x, y)$ and $\text{wt}(y + z) = d(y, z)$, the result follows. (This property is generally called the *Triangle Inequality*.)

When a function satisfies the four properties listed in Theorem 16.11, it is called a *distance function* or *metric*, and we call $(\mathbf{Z}_2^n, d)$ a *metric space*. Hence d (as given above) is often referred to as the *Hamming metric*. This metric is used in the following.

Definition 16.10 For $n, k \in \mathbf{Z}^+$ and $x \in \mathbf{Z}_2^n$, the *sphere* of radius k centered at x is defined as $S(x, k) = \{y \in \mathbf{Z}_2^n \mid d(x, y) \le k\}$.

EXAMPLE 16.23 For $n = 3$ and $x = 110 \in \mathbf{Z}_2^3$, $S(x, 1) = \{110, 010, 100, 111\}$ and $S(x, 2) = \{110, 010, 100, 111, 000, 101, 011\}$.

With these preliminaries in hand we turn now to the two major results of this section.

THEOREM 16.12 Let $E: W \to C$ be an encoding function with the set of messages $W \subseteq \mathbf{Z}_2^m$ and the set of code words $E(W) = C \subseteq \mathbf{Z}_2^n$, where $m < n$. If our objective is error detection, then for $k \in \mathbf{Z}^+$, we can detect all transmission errors of weight $\le k$ if and only if the minimum distance between code words is at least $k + 1$.

Proof: The set C is known to both the transmitter and the receiver, so if $w \in W$ is the message and $c = E(w)$ is transmitted, let $c \ne T(c) = r$. If the minimum distance between code words is at least $k + 1$, then the transmission of c can result in as many as k errors and r will not be listed in C. Hence we can detect all errors e where $\text{wt}(e) \le k$. Conversely, let c_1, c_2 be code words with $d(c_1, c_2) < k + 1$. Then $c_2 = c_1 + e$ where $\text{wt}(e) \le k$. If we send c_1 and $T(c_1) = c_2$, then we would feel that c_2 had been sent, thus failing to detect an error of weight $\le k$.

What can we say about error-correcting capability?

THEOREM 16.13 Let E, W, and C be as in Theorem 16.12. If our objective is error correction, then for $k \in \mathbf{Z}^+$, we can construct a decoding function $D: \mathbf{Z}_2^n \to W$ that corrects all transmission errors of weight $\le k$ if and only if the minimum distance between code words is at least $2k + 1$.

Proof: For $c \in C$, consider $S(c, k) = \{x \in \mathbf{Z}_2^n \mid d(c, x) \le k\}$. Define $D: \mathbf{Z}_2^n \to W$ as follows. If $r \in \mathbf{Z}_2^n$ and $r \in S(c, k)$ for some code word c, then $D(r) = w$ where $E(w) = c$. [Here c is the (unique) code word *nearest* to r.] If $r \notin S(c, k)$ for any $c \in C$, then we define $D(r) = w_0$, where w_0 is some arbitrary message that remains fixed once it is chosen. The only problem we could face here is that D might not be a function. This will happen if there is an element r in $\mathbf{Z}_2^n$ with r in both $S(c_1, k)$ and $S(c_2, k)$ for distinct code words c_1, c_2. But $r \in S(c_1, k) \Rightarrow d(c_1, r) \le k$, and $r \in S(c_2, k) \Rightarrow d(c_2, r) \le k$, so $d(c_1, c_2) \le d(c_1, r) + d(r, c_2) \le k + k < 2k + 1$. Consequently, if the minimum distance between code words is at least $2k + 1$, then D is a function, and it will decode all possible

received words, correcting any transmission error of weight $\leq k$. Conversely, if $c_1, c_2 \in C$ and $d(c_1, c_2) \leq 2k$, then c_2 can be obtained from c_1 by making at most $2k$ changes. Starting at code word c_1 we make approximately half (exactly, $\lfloor d(c_1, c_2)/2 \rfloor$) of these changes. This brings us to $r = c_1 + e_1$ with $\text{wt}(e_1) \leq k$. Continuing from r, we make the remaining changes to get to c_2 and find $r + e_2 = c_2$ with $\text{wt}(e_2) \leq k$. But then $r = c_2 + e_2$. Now with $c_1 + e_1 = r = c_2 + e_2$ and $\text{wt}(e_1)$, $\text{wt}(e_2) \leq k$, how can one decide on the code word from which r arises? This ambiguity results in a possible error of weight $\leq k$ that cannot be corrected.

16.24

With $W = \mathbf{Z}_2^2$ let $E\colon W \to \mathbf{Z}_2^6$ be given by

$$E(00) = 000000 \qquad E(10) = 101010 \qquad E(01) = 010101 \qquad E(11) = 111111.$$

Then the minimum distance between code words is 3, so we can correct all single errors. With

$$\begin{aligned} S(000000, 1) &= \{x \in \mathbf{Z}_2^6 | d(000000, x) \leq 1\} \\ &= \{000000, 100000, 010000, 001000, 000100, 000010, 000001\}, \end{aligned}$$

the decoding function $D\colon \mathbf{Z}_2^6 \to W$ gives $D(x) = 00$ for all $x \in S(000000, 1)$.

Similarly,

$$\begin{aligned} S(010101, 1) &= \{x \in \mathbf{Z}_2^6 | d(010101, x) \leq 1\} \\ &= \{010101, 110101, 000101, 011101, 010001, 010111, 010100\}, \end{aligned}$$

and here $D(x) = 01$ for each $x \in S(010101, 1)$. At this point our definition of D accounts for 14 of the elements in $\mathbf{Z}_2^6$. Continuing to define D for the 14 elements in $S(101010, 1)$ and $S(111111, 1)$ there remain 36 other elements to account for. We define $D(x) = 00$ (or any other message) for these 36 other elements and have a decoding function that will correct single errors.

Beware! There is a subtle point that needs to be made about Theorems 16.12 and 16.13. For example, if the minimum distance between code words is $2k + 1$ one may feel that we can detect all errors of weight $\leq 2k$ *and* correct all errors of weight $\leq k$. This is not necessarily true. That is, error detection and error correction need not take place at the same time and at the maximum levels. To see this, reconsider the (6, 2)-triple repetition code of Example 16.24. Here the encoding function $E\colon W(= \mathbf{Z}_2^2) \to \mathbf{Z}_2^6$ is given by $E(w_1w_2) = w_1w_2w_1w_2w_1w_2$ and the code comprises the four elements of $\mathbf{Z}_2^6$ in the range of E. Since the minimum distance between any two elements of $\mathbf{Z}_2^2$ is 1, it follows that the minimum distance between code words is 3 (as observed earlier in Example 16.24).

Now suppose that our major objective is error correction and that $r = 100000\ [\notin E(W)]$ is received. We see that $d(000000, r) = 1$, $d(101010, r) = 2$, $d(010101, r) = 4$, and $d(111111, r) = 5$. Consequently, we should choose to decode r as 000000, the unique code word nearest to r. Unfortunately, suppose that the actual message were 10 (with corresponding code word 101010), but we received $r = 100000$. Upon correcting r as 000000, we should then decode 000000 to get the incorrect message 00. And, in so doing, we have failed to detect an error of weight 2.

In this type of situation one can develop a scheme where a mixed strategy is used. Here both error correction *and* error detection may be carried out at some levels.

For $t \in \mathbf{N}$, if the received word is r and there is a unique code word c_1 such that $d(c_1, r) \leq t$, then we decode r as c_1. (*Note*: The case where $r = c_1$ is covered when $t = 0$.) If there exists a second code word c_2 such that $d(c_2, r) = d(c_1, r)$, or if $d(c, r) > t$ for all code words c, then an error is declared (and retransmission is generally requested). Using this scheme, if the minimum distance between code words is at least $2t + s + 1$, for $s \in \mathbf{N}$, then we can correct all errors of weight $\leq t$ and detect all errors with weights between $t + 1$ and $t + s$, inclusive.

When using this scheme for the (6, 2)-triple repetition code, our options include:

1) $t = 0$; $s = 2$: Here we can detect all errors of weight ≤ 2 but we have no error-correction capability.
2) $t = 1$; $s = 0$: Single errors are corrected here but there is no error-detecting capability.

If we use the (10, 2)-five-times repetition code, then the minimum distance is 5. Applying the above scheme in this case, our options now include:

1) $t = 0$; $s = 4$: Here we can detect all errors of weight ≤ 4 but we have no error-correction capability.
2) $t = 1$; $s = 2$: Now single errors are corrected and we can also detect all errors e, where $2 \leq \text{wt}(e) \leq 3$.
3) $t = 2$; $s = 0$: All errors of weight ≤ 2 are corrected but there is no error-detecting capability.

[For more on this, the interested reader should examine Chapter 4 of the text by S. Roman [24].]

16.7 The Parity-Check and Generator Matrices

In this section we introduce an example where the encoding and decoding functions are given by matrices over $\mathbf{Z}_2$. One of these matrices will help us to locate the *nearest* code word for a given received word. This will be especially helpful as the set C of code words grows larger.

EXAMPLE 16.25

Let

$$G = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}$$

be a 3×6 matrix over $\mathbf{Z}_2$. The first three columns of G form the 3×3 identity matrix I_3. Letting A denote the matrix formed from the last three columns of G, we write $G = [I_3|A]$ to denote its structure. The (partitioned) matrix G is called a *generator matrix*.

We use G to define an encoding function $E\colon \mathbf{Z}_2^3 \to \mathbf{Z}_2^6$ as follows. For $w \in \mathbf{Z}_2^3$, $E(w) = wG$ is the element in $\mathbf{Z}_2^6$ obtained by multiplying w, considered as a three-dimensional row vector, by the matrix G on its right. Unlike the results on matrix multiplication in Chapter 7, in the calculations here we have $1 + 1 = 0$, not $1 + 1 = 1$.

(Even if the set W of messages is not all of $\mathbf{Z}_2^3$, we'll assume that all of $\mathbf{Z}_2^3$ is encoded and that the transmitter and receiver will both know the real messages of importance and their corresponding code words.)

We find here, for example, that

$$E(110) = (110)G = [110]\begin{bmatrix}1&0&0&1&1&0\\0&1&0&0&1&1\\0&0&1&1&0&1\end{bmatrix} = [110101],$$

and

$$E(010) = (010)G = [010]\begin{bmatrix}1&0&0&1&1&0\\0&1&0&0&1&1\\0&0&1&1&0&1\end{bmatrix} = [010011].$$

Note that $E(110)$ can be obtained by adding the first two rows of G, whereas $E(010)$ is simply the second row of G.

The set of code words obtained by this method is

$$C = \{000000,\ 100110,\ 010011,\ 001101,\ 110101,\ 101011,\ 011110,\ 111000\} \subseteq \mathbf{Z}_2^6,$$

and one can recapture the corresponding message by simply dropping the last three components of the code word. In addition, the minimum distance between code words is 3, so we can detect errors of weight ≤ 2 or correct single errors. (We shall assume that multiple errors are rare and concentrate on error correction.)

For all $w = w_1w_2w_3 \in \mathbf{Z}_2^3$, $E(w) = w_1w_2w_3w_4w_5w_6 \in \mathbf{Z}_2^6$. Since

$$\begin{aligned}E(w) &= [w_1w_2w_3]\begin{bmatrix}1&0&0&1&1&0\\0&1&0&0&1&1\\0&0&1&1&0&1\end{bmatrix}\\ &= [w_1w_2w_3(w_1+w_3)(w_1+w_2)(w_2+w_3)],\end{aligned}$$

we have $w_4 = w_1 + w_3$, $w_5 = w_1 + w_2$, $w_6 = w_2 + w_3$, and these equations are called the *parity-check equations*. Since $w_i \in \mathbf{Z}_2$ for each $1 \leq i \leq 6$, it follows that $w_i = -w_i$ and so the equations can be rewritten as

$$\begin{aligned}w_1 \quad\quad + w_3 + w_4 \quad\quad &= 0\\ w_1 + w_2 \quad\quad + w_5 \quad\quad &= 0\\ w_2 + w_3 \quad\quad + w_6 &= 0.\end{aligned}$$

Thus we find that

$$\begin{bmatrix}1&0&1&1&0&0\\1&1&0&0&1&0\\0&1&1&0&0&1\end{bmatrix}\begin{bmatrix}w_1\\w_2\\w_3\\w_4\\w_5\\w_6\end{bmatrix} = H\cdot(E(w))^{\mathrm{tr}} = \begin{bmatrix}0\\0\\0\end{bmatrix},$$

where $(E(w))^{\mathrm{tr}}$ denotes the transpose of $E(w)$. Consequently, if $r = r_1r_2\cdots r_6 \in \mathbf{Z}_2^6$, we can identify r as a code word if and only if

$$H\cdot r^{\mathrm{tr}} = \begin{bmatrix}0\\0\\0\end{bmatrix}.$$

Writing $H = [B|I_3]$, we notice that if the rows and columns of B are interchanged, then we get A. Hence $B = A^{\mathrm{tr}}$.

From the theory developed earlier on error correction, because the minimum distance between the code words of this example is 3, we should be able to develop a decoding function that corrects single errors.

Suppose we receive $r = 110110$. We want to find the code word c that is the *nearest neighbor* of r. If there is a long list of code words against which to check r, we would be better off to first examine $H\cdot r^{\mathrm{tr}}$, which is called the *syndrome* of r. Here

$$H\cdot r^{\mathrm{tr}} = \begin{bmatrix}1&0&1&1&0&0\\1&1&0&0&1&0\\0&1&1&0&0&1\end{bmatrix}\begin{bmatrix}1\\1\\0\\1\\1\\0\end{bmatrix} = \begin{bmatrix}0\\1\\1\end{bmatrix},$$

so r is not a code word. Hence we at least detect an error. Looking back at the list of code words, we see that $d(100110, r) = 1$. For all other $c \in C$, $d(r, c) \geq 2$. Writing $r = c + e = 100110 + 010000$, we find that the transmission error (of weight 1) occurs in the second component of r. Is it just a coincidence that the syndrome $H\cdot r^{\mathrm{tr}}$ produced the second column of H? If not, then we can use this result in order to realize that if a single transmission error occurred, it took place at the second component. Changing the second component of r, we get c; the message w comprises the first three components of c.

Let $r = c + e$, where c is a code word and e is an error pattern of weight 1. Suppose that 1 is in the ith component of e, where $1 \leq i \leq 6$. Then

$$H\cdot r^{\mathrm{tr}} = H\cdot(c+e)^{\mathrm{tr}} = H\cdot(c^{\mathrm{tr}} + e^{\mathrm{tr}}) = H\cdot c^{\mathrm{tr}} + H\cdot e^{\mathrm{tr}}.$$

With c a code word, it follows that $H\cdot c^{\mathrm{tr}} = \mathbf{0}$, so $H\cdot r^{\mathrm{tr}} = H\cdot e^{\mathrm{tr}} = i$th column of matrix H. Thus c and r differ only in the ith component, and we can determine c by simply changing the ith component of r.

Since we are primarily concerned with transmissions where multiple errors are rare, this technique is of definite value. If we ask for more, however, we find ourselves expecting too much.

Suppose that we receive $r = 000111$. Computing the syndrome

$$H\cdot r^{\mathrm{tr}} = \begin{bmatrix}1&0&1&1&0&0\\1&1&0&0&1&0\\0&1&1&0&0&1\end{bmatrix}\begin{bmatrix}0\\0\\0\\1\\1\\1\end{bmatrix} = \begin{bmatrix}1\\1\\1\end{bmatrix},$$

we obtain a result that is not one of the columns of H. Yet $H\cdot r^{\mathrm{tr}}$ can be obtained as the sum of two columns from H. If $H\cdot r^{\mathrm{tr}}$ came from the first and sixth columns of H, correcting these components in r results in the code word 100110. If we sum the third and fifth columns of H to get this syndrome, upon changing the third and fifth components of r we get a second code word, 001101. So we cannot expect H to correct multiple errors. This is no surprise since the minimum distance between code words is 3.

We summarize the results of Example 16.25 for the general situation. For $m, n \in \mathbf{Z}^+$ with $m < n$, the encoding function $E\colon \mathbf{Z}_2^m \to \mathbf{Z}_2^n$ is given by an $m \times n$ matrix G over $\mathbf{Z}_2$. This matrix G is called the generator matrix for the code and has the form $[I_m|\ A]$, where