In many applications, errors are not randomly distributed Instead, they occur in bursts For example, in a CD, a scratch introduces errors in many adjacent bits A burst of solar energy could have a similar effect on communications from a spacecraft Reed-Solomon codes are useful in such situations

For example, suppose we take $\mathbf{F} = GF(2^8)$ The elements of $\mathbf{F}$ are represented as bytes of eight bits each, as in Section 3 10 We have $n = 2^8 - 1 = 255$ Let $d = 33$ The codewords are then vectors consisting of 255 bytes There are 222 information bytes and 33 check bytes These codewords are sent as strings of $8 \times 255 = 2040$ binary bits Disturbances in the transmission will corrupt some of these bits However, in the case of bursts, these bits will often be in a small region of the transmitted string If, for example, the corrupted bits all lie within a string of 121 $(= 15 \times 8 + 1)$ consecutive bits, there can be errors in at most 16 bytes Therefore, these errors can be corrected (because $16 < d/2$) On the other hand, if there were 121 bit errors randomly distributed through the string of 2040 bits, numerous bytes would be corrupted, and correct decoding would not be possible. Therefore, the choice of code depends on the type of errors that are expected.

Code Based Cryptography(CBC)
Non-Commutative Cryptography (NCC)
Post-Quantum Cryptography(PQC)

Integer Factoring Problem(IFP)
Discrete Logarithm Problem(DLP)
Nearest Codeword Problem(NCP)

16.10 The McEliece Cryptosystem

In this book, we have mostly described cryptographic systems that are based on number theoretic principles There are many other cryptosystems that are based on other complex problems Here we present one based on the difficulty of finding the nearest codeword for a linear binary code

The idea is simple Suppose you have a binary string of length 1024 that has 50 errors There are $\binom{1024}{50} \approx 3 \times 10^{85}$ possible locations for these errors, so an exhaustive search that tries all possibilities is infeasible Suppose, however, that you have an efficient decoding algorithm that is unknown to anyone else Then only you can correct these errors and find the corrected string. McEliece showed how to use this to obtain a public key cryptosystem.

Bob chooses G to be the generating matrix for an (n, k) linear error correcting code C with $d(C) = d$ He chooses S to be a $k \times k$ matrix that is invertible mod 2 and lets P be an $n \times n$ permutation matrix, which means that P has exactly one 1 in every row and in every column, with all the other entries being 0 Define

k*k, k*n, n*n

C(G)与C(G1)等价

$$G_1 = SGP$$

The matrix G_1 is the public key for the cryptosystem Bob keeps S, G, P secret.

In order for Alice to send Bob a message x, she generates a random binary string e of length n that has weight t She forms the ciphertext by

computing

$$y \equiv xG_1 + e \pmod 2$$

Bob decrypts y as follows

1 Calculate $y_1 \equiv yP^{-1}$ (Since P is a permutation matrix, $e_1 = eP^{-1}$ is still a binary string of weight t We have $y_1 \equiv xSG + e_1$)

2 Apply the error decoder for the code C to y_1 to correct the "error' and obtain the codeword x_1 closest to y_1

3 Compute x_0 such that $x_0G \equiv x_1$ (in the examples we have considered, x_0 is simply the first k bits of x_1).

4 Compute $x \equiv x_0S^{-1}$.

The security of the system lies in the difficulty of decoding y_1 to obtain x_1 There is a little security built into the system by S; however, once a decoding algorithm is known for the code generated by GP, a chosen plaintext attack allows one to solve for the matrix S (as in the Hill cipher).

To make decoding difficult, $d(C)$ should be chosen to be large McEliece suggested using a $[1024, 512, 101]$ Goppa code. The **Goppa codes** have parameters of the form $n = 2^m$, $d = 2t + 1$, $k = n - mt$ For example, taking $m = 10$ and $t = 50$ yields the $[1024, 524, 101]$ code just mentioned It can correct up to 50 errors For given values of m and t, there are in fact many inequivalent Goppa codes with these parameters. We will not discuss these codes here except to mention that they have an efficient decoding algorithm and therefore can be used to correct errors quickly.

Example. Consider the matrix

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix},$$

which is the generator matrix for the $[7, 4]$ Hamming code Suppose Alice wishes to send a message

$$m = (1, 0, 1, 1)$$

to Bob. In order to do so, Bob must create an invertible matrix S and a random permutation matrix P that he will keep secret. If Bob chooses

$$S = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

and

$$P = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

Using these, Bob generates the public encryption matrix

$$G_1 = \begin{pmatrix} 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 \end{pmatrix}.$$

In order to encrypt, Alice generates her own random error vector e and calculates the ciphertext $y = xG_1 + e$ In the case of a Hamming code the error vector has weight 1 Suppose Alice chooses

$$e = (0, 1, 0, 0, 0, 0, 0)$$

Then

$$y = (0, 0, 0, 1, 1, 0, 0).$$

Bob decrypts by first calculating

$$y_1 = yP^{-1} = (0, 0, 1, 0, 0.0, 1)$$

Calculating the syndrome of y_1 by applying the parity check matrix H and changing the corresponding bit yields

$$x_1 = (0, 0, 1.0, 0, 1, 1).$$

Bob next forms a vector x_0 such that $x_0G = x_1$, which can be done by extracting the first four components of x_1, that is,

$$x_0 = (0, 0, 1, 0)$$

Bob decrypts by calculating

$$x = x_0S^{-1} = (1, 0, 1, 1),$$

which is the original plaintext message. ∎

The McEliece system seems to be reasonably secure. For a discussion of its security, see [Chabaud]. A disadvantage of the system compared to RSA, for example, is that the size of the public key G_1 is rather large.

16.11 Other Topics

The field of error correcting codes is a vast subject that is explored by both the mathematical community and the engineering community. In this chapter we have only touched upon a select handful of the concepts of this field. There are many other areas of error correcting codes that we have not discussed

Perhaps most notable of these is the study of convolutional codes In this chapter we have entirely focused on block codes, where typically the data are segmented into blocks of a fixed length k and mapped into codewords of a fixed length n. However, in many applications, the data are produced in a continuous fashion, and it is better to map the stream of data into a stream of coded symbols. For example, such codes have the advantage of not requiring the delay needed to observe an entire block of symbols before encoding or decoding.

Another topic that is very important in the study of error correcting codes is that of efficient decoding. In the case of linear codes, we presented syndrome decoding, which is more efficient than performing a search for the nearest codeword. However, for large linear codes, syndrome decoding is still too inefficient to be practical. When BCH and Reed-Solomon codes were introduced, the decoding schemes that were originally presented were impractical for decoding more than a few errors. Later, Berlekamp and Massey provided an efficient approach to decoding BCH and Reed-Solomon codes. There is still a lot of research being done on this topic. We direct the reader to the books [Lin-Costello], [Wicker], [Gallager], and [Berlekamp] for further discussion on the subject of decoding.

We have also focused entirely on bit or symbol errors. However, in modern computer networks, the types of errors that occur are not simply bit or symbol errors but also the complete loss of segments of data. For example, on the Internet, data are transferred over the network in chunks called packets. Due to congestion at various locations on the network, such as routers and switches, packets might be dropped and never reach their intended recipient. In this case, the recipient might notify the sender, requesting a packet to be resent. Protocols such as the Transmission Control Protocol (TCP) provide mechanisms for retransmitting lost packets

When performing cryptography, it is critical to use a combination of many different types of error control techniques to assure reliable delivery of encrypted messages; otherwise, the receiver might not be able decrypt the messages that were sent.

Finally, we mention that coding theory has strong connections with various problems in mathematics such as finding dense packings of high-dimensional spheres. For more on this, see [Thompson]

16.12 Exercises

1. Two codewords were sent using the Hamming $[7,4]$ code and were received as 0100111 and 0101010. Each one contains at most one error. Correct the errors. Also, determine the 4-bit messages that were multiplied by the matrix G to obtain the codewords.

2. An ISBN number is incorrectly written as 0-13-116093-8. Show that this is not a correct ISBN number. Find two different valid ISBN numbers such that an error in one digit would give this number. This shows that ISBN cannot correct errors

3. The following is a parity check matrix for a binary $[n,k]$ code C

$$\begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

(a) Find n and k
(b) Find the generator matrix for C.
(c) List the codewords in C.
(d) What is the code rate for C?

4. Let $C = \{(0,0,0),(1,1,1)\}$ be a binary repetition code
(a) Find a parity check matrix for C.
(b) List the cosets and coset leaders for C
(c) Find the syndrome for each coset
(d) Use the syndrome decoding method to decode the message $(1,1,0)$.

5. Let C be the binary code $\{(0,0,0),(1,1,0),(1,0,1),(0,1,1)\}$
(a) Show that C is not linear
(b) What is $d(C)$? (Since C is not linear, this cannot be found by calculating the minimum weight.)
(c) Show that C satisfies the Singleton bound with equality

6. Show that the weight function (on $\mathbf{F}^n$) satisfies the triangle inequality. $wt(u+v) \leq wt(u) + wt(v)$.

7. Show that $A_q(n,n) = q$, where $A_q(n,d)$ is the function defined in Section 16 3

8. Let C be the repetition code of length n Show that $C^\perp$ is the parity check code of length n (This is true for arbitrary $\mathbf{F}$.)

9. Let C be a linear code and let $u+C$ and $v+C$ be cosets of C. Show that $u+C=v+C$ if and only if $u-v\in C$. (*Hint*: To show $u+C=v+C$, it suffices to show that $u+\mathrm{c}\in v+C$ for every $\mathrm{c}\in C$, and that $v+\mathrm{c}\in u+C$ for every $\mathrm{c}\in C$. To show the opposite implication, use the fact that $u\in u+C$.)

10. Show that if C is a self-dual $[n,k,d]$ code, then n must be even.

11. Show that $g(X)=1+X+X^2+\cdots+X^{n-1}$ is the generating polynomial for the $[n,1]$ repetition code. (This is true for arbitrary $\mathbf{F}$.)

12. Let $g(X)=1+X+X^3$ be a polynomial with coefficients in $\mathbf{Z}_2$.
(a) Show that $g(X)$ is a factor of X^7-1 in $\mathbf{Z}_2[X]$.
(b) The polynomial $g(X)$ is the generating polynomial for a cyclic code $[7,4]$ code C. Find the generating matrix for C.
(c) Find a parity check matrix H for C.
(d) Show that $G'H^T=0$, where

$$G'=\begin{pmatrix}1&1&0&1&0&0&0\\0&1&1&0&1&0&0\\0&0&1&1&0&1&0\\0&1&1&1&0&0&1\end{pmatrix}.$$

(e) Show that the rows of G' generate C.
(f) Show that a permutation of the columns of G' gives the generating matrix for the Hamming $[7,4]$ code, and therefore these two codes are equivalent.

13. Let C be the cyclic binary code of length 4 with generating polynomial $g(X)=X^2+1$. Which of the following polynomials correspond to elements of C?

$$f_1(X)=1+X+X^3,\quad f_2(X)=1+X+X^2+X^3,\quad f_3(X)=X^2+X^3.$$

14. Let $g(X)$ be the generating polynomial for a cyclic code C of length n, and let $g(X)h(X)=X^n-1$. Write $h(X)=b_0+b_1X+\cdots+X^\ell$. Show that the dual code $C^\perp$ is cyclic with generating polynomial $\tilde{h}_r(X)=(1/b_0)(1+b_{\ell-1}X+\cdots+b_1X^{\ell-1}+b_0X^\ell)$. (The factor $1/b_0$ is included to make the highest nonzero coefficient be 1.)

15. **(a)** Let C be a binary repetition code of odd length n (that is, C contains two vectors, one with all 0's and one with all 1's). Show that C is perfect. (*Hint*: Show that every vector lies in exactly one of the two spheres of radius $(n-1)/2$.)
(b) Use (a) to show that if n is odd then $\sum_{j=0}^{(n-1)/2}\binom{n}{j}=2^{n-1}$. (This can also