Preface

It is our privilege and pleasure to welcome all our readers to the dynamic world of cryptography, information theory and error correction. Both authors have considerable industrial experience in the field. Also, on the academic side Dr. Bruen has been a long-time editor of leading research journals such as "Designs, Codes and Cryptography". Prior to his appointment in Calgary he worked in mathematical biology at Los Alamos. The book is an outgrowth both of presentations to industry groups and of a lecture course at the University of Calgary. The course was for undergraduate and graduate students in Computer Science, Engineering and Mathematics.

In addition to the academic topics in that course, we also include material relating to our industrial consulting work and experience in writing patents on the topics in the title of the book. In particular we describe revolutionary new algorithms in chapter 24 for hash functions and symmetric cryptography including quantum cryptography. These have been patented and have already made their way into industry.

This book can be read at many different levels. For example, it can be used as a reference or a text for courses in any of the three subjects or for a combined course. To this end we have included over three hundred worked examples and problems, with answers or solutions as needed. But we were determined to make the work highly accessible to the general reader as well. We hope that the exposition fulfills this goal. Large sections of this book have been written in such a way that little is required in the way of mathematical background. In places this was difficult to do but we believe that the effort has been worthwhile.

The three topics become more and more entwined as science and technology develop. In our opinion, the time when the three topics can be treated in isolation is rapidly drawing to a close. For example, if you search the internet for cryptographic information it is more and more likely that you will run up against terms such as entropy, CRC checksums, random number generators and the like. [Digressing: the main undergraduate course in computer science which is concerned with data structures — and we have all taught it — covers Huffman codes and compression at length but the word entropy is never mentioned. This is a shame].

Thus it seemed quite appropriate to us to try to write a complete but highly accessible

account of the three subjects stressing, above all, their interconnections and their unity. These interconnections can be hidden if one relies only on separate accounts of the three subjects. In addition, as part of information theory, we discuss some potential applications in cell biology. In the last chapter we present some new, exciting algorithms which combine all three of the subjects.

This is not the first time that a book combining the three subjects has been attempted. Several very good recent books, specializing in cryptography, have a few chapters on the other subjects. But our goal was to give a full in-depth account. We should mention that other books have handled nicely two of the three topics. A splendid book, published in 1988 by Dominic Welsh gives an account of all three of the subjects. However, a lot has happened since 1988. Also, our focus, emphasis and level of detail is different.

Let us briefly explain the 3 subjects.

Cryptography. This is an ancient subject concerned with the secret transmission of messages between two parties denoted by A and B. This could be done if A, B shared a secret language, say, not known to outsiders. More generally they can communicate in secret by sharing a common secret "key". Then A uses the key to scramble the message to B, who unscrambles the message with a copy of the same key that is owned by A. We may think of military commanders sending secret messages to each other or home movie providers sending movies to authorized customers. Apart from secrecy there are also crucial questions in cryptography involving authentication and identification.

Information Theory. This subject, also known as Shannon Theory after Claude Shannon, the late American mathematician and engineer, gives precise mathematical meaning to the term "information". This leads to answers to such questions as the following:

How much compression of data can be carried out without losing any information?

What is the maximum amount of information that can be transmitted over a noisy channel?

This fundamental question is answered precisely in Shannon's famous channel capacity theorem which was discovered around 1948.

Error Correction. We introduce redundancy ["good redundancy"] for the transmission of messages, as opposed to the "bad redundancy" which was banished using compression. In this way we try to ensure that the receiver decodes accurately within the bounds of the Shannon capacity theorem mentioned in the previous section. The wonderful pictures of far-away planets, that have recently been made available, are just one example of what error-correcting codes can do. With a modern modem we can both compress as well as encode and decode to any required degree of accuracy.

Interconnections. These are spelled out in detail in the text but let us give a few short informal connections. How secure is your cryptographic password? It depends on how hard it is to guess it, i.e. it depends on its entropy as measured in Shannon bits. We then need information theory to properly discuss this.

In cryptography, A is sending information in secret to B, but what exactly is information and how is it measured? Again we need information theory.

Suppose that A is sending a secret key K over a channel to B in order to encrypt, at some future date, a secret message M with K and transmit it to B. Now, a basic property of K is this. If the transmission of K is off by even one big then B will end up with a message that is completely different from the intended message M . The bottom line is that a transmission error could be catastrophic. The best way to guard against this is for A to use robust error-correction when sending the cryptographic key K to B.

The great Claude Shannon made the following fundamental point. In error-correction, the receiver B is trying to correctly decode what the transmitter A has sent to B over a "noisy channel". Compare this to the cryptographic situation where A is sending secret messages to B. They must contend with the eavesdropper — the evil Eve — who is listening in. We can think of Eve as receiving a "noisy" version of M and trying to decipher, or decode M. We are back to coding theory. [Parenthetically, we mention that Shannon designed an interesting theory of the stock market by regarding the market as a very noisy channel!].

We must point out that this point of view of Shannon is extremely useful and not just as a formal device. We drive the point home with several problems in Chapter 16 where the analogy becomes quite striking. Moreover, in Chapter 24, A and B and Eve may have the same information to start out with, yet A and B have to come up with a way of beating Eve and publicly generating a secret key using a technique known as "Privacy Amplification".

Here is yet another basic interconnection. Random numbers and pseudo-random numbers are the work-horses of cryptography, especially symmetric cryptography. One of the best ways of generating them is with shift-registers. In fact, as is pointed out in Schneier [Sch96], "stream ciphers based on shift registers have been the workhorse of military cryptography since the beginning of electronics". But shift registers are central in information theory as they are great proving-grounds [or grave-yards] for questions on entropy. To understand entropy you have to confront shift registers. But — and here is the astonishing part — these shift registers, over any field, correspond exactly to cyclic linear codes which are at the heart of error-correction. For the expert, Reed–Solomon codes, and not just their error-correction, are merely special kinds of shift registers in disguise!.

We move on now to a more conventional-type preface and address some standard questions.

Intended Readership. This is a book for everyone and can be used at many different levels. We are writing for many different kinds of readers.

1. All-rounders or renaissance types who have taken some mathematics or computer science or engineering [or none of the above] and who want to find out about these topics and have some fun.
2. Undergraduates or graduate students in mathematics, computer science or engineering.
3. Instructors of algebra and linear algebra who would like some real life practical applications in their courses, such as shift registers.
4. Biologists who may be interested in our discussions of such topics as biological compression and the channel capacity corresponding to the genetic code.
5. IT workers, venture capitalists and others who want an overview of the basics.
6. Academics looking for a good source of important (and doable) research problems.
7. Philosophers and historians of science who want to move on from quantum theory and relativity to a new, practical area which also, incidentally, has strong connections to quantum mechanics.

Rewards for Readers. If you make a good effort at understanding this book and working out some of the problems you will be well rewarded. This book covers everything you need. In particular, you will elevate your skills and mathematical maturity to a new level. You will also have an excellent background — better than that of most practitioners — in these areas. You will be ready to think about a career in cryptography or codes or even information theory. The market, especially in such areas as data compression is hot. You will be very well-placed for advanced work in cryptography, error-correction or information theory.

Our Goals. We want to help develop your skills and inspire you to new heights. Let this book be your inspiration. Master it and then get out and write those patents!

Possible Courses Using this Book. There is more than enough material for a stand-alone course at the undergraduate or graduate levels, in any of the three areas. The extensive list of problems and worked examples will be a big help. For those few chapters that don't have problems there are opportunities for many fun group-projects geared towards reporting on patents, publications etc. We would recommend some "poaching" among the three parts of the book in such a course. A one year combined course would also work well.

A Course for Non-Specialists. Most of Part I, apart from the Chapter on elliptic curves, requires very little mathematical background but covers a lot of ground in cryptography.In information theory, we highly recommend Chapter 10 which gives a panorama of information theory and interesting related topics such as the "MBA problem" on weighings. The chapter

on topics related to the genetic code does not require much background, and should be of considerable interest. We also recommend Chapter 18 introducing coding theory. Chapter 20 tells the amazing story of how the famous perfect Golay code G_{11} was first published in a Finnish soccer magazine in connection with the football pools in that country. Chapter 24 describes what appears to be a breakthrough in symmetric cryptography, error correction, and hash functions. We highly recommend it!

Level, Mathematical Style, Proofs, Exercises. We have made a considerable effort to ensure that the chapters are as accessible as possible. In terms of style, our motto, which is the opposite of many mathematicians and engineers, is this: "Never use a symbol if you can get away with a word."

What about proofs? It really depends. If the proof enhances the ideas we try to present it. Also, some results, such as the Shannon source-coding result are so astonishing that we have to give the details. However, in the case of the noisy channel theorem we have a different approach. From teaching, we found it considerably more effective to give five or six different approaches rather than to just give the standard official proof.

This book was not written just for theoreticians. Much of our time was spent in designing good problems and solution. We urge our readers to take advantage of them.

Mathematical Prerequisites. Honestly? We try to cover everything "on the fly" along with one special chapter on specialized topics but here is a short summary of what we need.

Calculus: a small amount having to with the concavity of a graph [second derivative] and function maxima [first derivative, end points].

Linear algebra: Multiplying matrices, subspaces, invertibility and determinants.

Elementary probability and statistics: Mean and variance, Bernoulli trials, the normal curve, law of large numbers.

Algebra: A small amount of material on groups, finite fields, modular arithmetic.

Here and there we go over the top. For example, a bit of Fourier analysis for the Shannon sampling theorem is needed. But generally speaking, the above list covers most of the material and we do discuss the needed background as we go along.

What's New. Most of the Chapter have a "New, Noteworthy" heading where we try to summarize such matters. However, here is a brief summary of "what's new" in the book. The topics are listed in no particular order.

- An in-depth integrated discussion of cryptography, information theory and error-correction emphasizing their interconnections, including new, clear, accessible proofs of major results, along with new results.

- A discussion of RSA that clears up several issues and shows how, for example, a given encryption index may have several decryption indices: Also,an indication of a possible new attack on RSA.
- A study of potential applications of information theory in cellular biology.
- An overview of important practical considerations in modern cryptography and communication theory.
- A whole new treatment of "perfect secrecy", including a refutation of the standard assertion concerning the equivalence of perfect secrecy and the one-time pad, together with a proof of the equivalence of perfect secrecy and latin squares.
- A highly accessible summary of information theory and its applications for non-specialists for non-specialists.
- A detailed look at hash functions from the point of view of linear codes.
- A detailed discussion of shift registers in cryptography, information theory and error correction including several new results and their application to the Berlekamp–Massey theory of Reed–Solomon and BCH decoding.
- A clarification of several points of confusion in the literature relating to security.
- A presentation of five different approaches to Shannon's noisy channel theorem.
- A detailed discussion of the sampling theorem and Shannon's fundamental band-limited capacity formula to the effect that $C = B\log(1 + \frac{S}{N})$, using precise statistical and geometrical techniques.
- A look at some of the history of cryptography and coding theory including a brief biography of Claude Shannon and an account of the original discovery of the Golay code in a Finnish soccer-pools magazine.
- A description of invariant theory and combinatorics applied to coding theory with particular reference to "the computer algebra theorem of the twentieth century" i.e. the nonexistence of a plane of order 10 and related work of one of the authors.
- Connections between MDS codes, secret-sharing schemes, Bruck Nets and Euler's "famous problem of the 36 officers".
- A brief description of research work due to the author and two co-authors solving, in the main, the fifty year old problem of finding the longest MDS code.
- A streamlined approach to Reed–Solomon codes via MDS codes.

Chapter 3

RSA, Key Searches, SSL, and Encrypting Email

Goals, Discussion This chapter is pivotal, but no mathematical background is required. We avoid making essential use of number theory in the text although it can be used to shorten the calculations. By the end of this chapter the reader will be well-versed in the main public key algorithm, namely RSA, as well as its applications to e-commerce with SSL and to encrypting email[PGP and GPG]. The reader will also be clear on the two kinds of cryptography (public key and symmetric) and will know about searching key-spaces. Some cryptographic attacks,both mathematical and real-world are discussed here and in Chapter 7.

One important difference between the two kinds of encryption is this: If an eavesdropper Eve immediately tries to guess the secret message she can verify whether or not the guess is correct whenever a public key system such as RSA is being used. However, this is not the case with symmetric encryption.

We discuss in detail one public key algorithm, namely the RSA algorithm, named after its inventors (or co-inventors) Rivest, Shamir, and Adelman. We show how anyone can learn to carry out RSA on a calculator.

Let us briefly explain the idea. Alice wants to send a secret message to Bob. Bob has chosen a number N and another number e (for encryption). Very often e is three or seventeen. The pair $[N, e]$ represents *Bob's public key* and is listed in a "public key directory." Alice represents the secret message as a number M lying between 1 and $N-1$. To encrypt or scramble the message M, Alice multiplies M by itself e times, gets the remainder after dividing by N, and transmits the result to Bob. The result is called the cipher text C. An eavesdropper, noting C, realizes that the message itself must be equal to the e^{th} root of C, or $C+N$, or $C+2N$, or $C+rN$ for some unknown r. Eve (the

eavesdropper) cannot find M as there are too many values of r to try. *It is a remarkable fact that if Eve can guess a single value of r, say $r = r_0$, such that the e^{th} root of $C + rN$ is a whole number lying between 1 and $N - 1$, then this whole number must indeed be M.* Moreover, it can be shown that if for any positive integer r the e^{th} root of $C+rN$ is a whole number v, then the remainder of v upon division by N must be M (see Chapter 19).

The recipient Bob, however, can calculate M immediately from a formula involving his private key consisting of a "decryption index" d along with two prime numbers p, q. The reason is that N is the product of p and q. Bob knows p and q. Anybody else, even knowing N, cannot determine what the factors p, q are in a reasonable amount of time. We can think of the public key as being located in the transmitter and a private (secret) key as being located in the receiver.

Eve can try guessing the message without knowing d by guessing r_0. Alternatively Eve can try guessing p and q from which she can calculate d. In other words Eve can try to guess the private key and then determine the message.

We detail some potential weaknesses with public key algorithms such as RSA. However, this algorithm is still the main public key algorithm. Its security, when carefully implemented, seems to still be strong after 25 years of constant use.

One of the most useful facts in cryptography is that in a **brute-force attack** on a key-space (one where we try all possible keys), the correct key is likely to be found after trying about half the total number of keys. Proofs of this statement seem to be rare. In this chapter we provide a short, simple proof of this fact.

Of course, only in special situations, such as when dealing with public key algorithms, can we check, with certainty, that we have the right key.

New, Noteworthy The encryption exponent e must be chosen to have no factors in common with $p-1$ and no factors in common with $q-1$. One reason for assuming this is so that d can then be calculated. However, a more basic reason is that this condition must be satisfied in order that two different messages get two different encryptions. This is discussed in the problems. Another interesting fact is that for a given $[N, e]$, **the decryption index need not be unique!** We provide several examples. This is important because some attacks on RSA are possible if d is small; we refer the reder to Chapter 7. So, if d is not unique, this makes it more difficult to guard against this attack.

What we mean by "not unique" is that there may be more than one value of d such that the remainder of C^d, on division by N, is M. The reason is that instead of working with $(p-1)(q-1)$, we can work with any number t that is divisible by $p-1$ and $q-1$, as explained in the algorithm description. It is always possible to find $t < (p-1)(q-1)$ so that the calculations are simplified, and we get a shortcut even if the resulting d is the same.

We give new mathematical insights on symmetric encryption and on such questions as the security of algorithms such as DES. The answer is that it all depends on the context and

Chapter 7

General and Mathematical Attacks in Cryptography

Goals, Discussion Some of the classical cryptanalytic techniques were already introduced in Chapter 2 for what are often referred to as "pencil and paper" ciphers. In Chapter 3 we also discussed several attacks on various implementations of RSA. Here we introduce several techniques, mathematical and otherwise, developed for the breaking of modern ciphers and other components of crypto-systems used in real-life applications and protocols. These are in addition to the attacks on insecure implementations of RSA discussed in various problems in Chapter 3.

New, Noteworthy In addition to giving a comprehensive overview of the most common attacks on privacy, we detail how the combination of a result described in Chapter 3 with the known attack on low decryption exponent suggest a possible new attack on RSA.

7.1 Cryptanalysis

Technically speaking, we can define **cryptanalysis** (or cryptanalytics or cryptoanalysis) as the art and science of solving unknown codes and ciphers. Cryptanalysts try to break the codes and ciphers created and used by cryptographers. This can sometimes be achieved either by obtaining the key or by directly obtaining the message. By extension, the art of exploiting weaknesses in protocols used to communicate secure information (authentication, key-management, software/hardware defects, etc.) also falls within the bounds of cryptanalysis. Practitioners often refer to the cryptanalytic process as "breaking the crypto-system".

In general, cryptanalysts are faced with the task of first determining the language being used for the communication, the general type of cipher or code being used, the specific

Chapter 9

Information Theory and Its Applications

Goals, Discussion We present a fairly complete but accessible survey of the subject involving ideas from mathematics, physics, and engineering. Not much mathematical background is required.

New, Noteworthy By discussing two of the main results we bridge the gap in Section 9.9 between abstract units of information, known as Shannon bits, and regular bits (= binary digits). We also explain a connection between weighing problems and information theory in Section 9.8. We show in Section 9.4 how information theory can be used in a fundamental way in cryptography. Connections with physics are explored.

9.1 Axioms, Physics, Computation

Information theory can be approached from many different points of view. It has many strands. In some treatments, it is an arcane mathematical subject concerned with axioms, measure theory, and abstract probability theories. In other treatments, it is described in terms of formulae that are the underpinnings of practical questions in modern communication theory in science and engineering. To others it is a subject that is inextricably linked with physics, notably statistical physics, heat, energy and the theory of computation as in Feynmann [Fey99]. Another approach is to tie the subject to complexity and randomness as developed by Solomonoff, Kolmogorov, Chaitin, and others. In this chapter, we will try to touch on these ideas.

As pointed out in Chapter 3, the subject has been dominated by Claude Shannon for over fifty years, but there have been several new applications and developments. For example,

information theory is being used in an important way in molecular biology and genetics. In nanotechnology, the parallel theory of quantum information is coming to the fore (Nielsen and Chung [NC]). In this book, we present, in Part III, a new application of information theory using classic physics, to cryptography. This application has recently made its way to the market place and to industrial applications.

Basically, information theory has to do with converting knowledge about probabilities to hard "information," suitably defined, involving units called **Shannon bits**. Ideas of randomness and redundancy follow closely. Rather than launching into definitions we first skirmish, informally, with some ideas.

9.2 Entropy

We start off with numbers. Any positive integer such as 43 can be expressed uniquely as $3 \times 10^0 + 4 \times 10^1 = 43$ using the powers of ten but we can use any other base, or number system, such as binary. For example, in binary, 43 can be written as the binary string of length 6 given by 101011. To explain this, we have $43 = 3 \times 10^0 + 4 \times 10^1$ if we read from the right. Also, reading from the right, $43 = 1 \times 2^0 + 1 \times 2^1 + 0 \times 2^2 + 1 \times 2^3 + 0 \times 2^4 + 1 \times 2^5$.

Now, suppose we are told that a given number is written as a binary string of length 6, and we are asked to guess that number. What are our chances of success? The total number of possibilities is 64, ranging from the number 0 to the number 63. To see this, the rightmost slot can be filled in two ways, and similarly for all the other slots. Thus we have $2 \times 2 \times 2 \times 2 \times 2 \times 2 = 64$ possibilities, ranging from 0 to 63, since $2^6 = 64$. So our probability of success in guessing is $\frac{1}{64}$. In general, if the binary string is of length n, i.e., has n bits, each of which is chosen independently, randomly deciding between 0 and 1 at each stage, there would be 2^n possible numbers from which to choose. Thus n binary digits can indicate any one of 2^n possible numbers. In this case, the **uncertainty** or **entropy** of the string is defined to be n because $n = \log(2^n)$. In this book, log invariably means log to the base 2. In general, we can say that, for a binary string,

$$entropy = \log(number\ of\ possible\ strings) \tag{9.1}$$

Getting back to the case $n = 6$, the number of possible numbers or outcomes would be reduced if we had some extra algebraic information. For example, if we knew that the six bits added up to zero (addition in binary) then any one of the six bits, for example, the sixth bit, is **redundant**. It can be calculated from the other five bits and thus causes no extra uncertainty. In this case, the string of length 6 is **less random** than before. The entropy now is just 5. So we have three principles:

Chapter 11

Source Coding, Data Compression, Redundancy

Goals, Discussion In this chapter we discuss source coding. At this point we are not yet transmitting data over a possibly noisy channel. Instead, we are mainly involved with formatting source words from a source into binary strings that can then be transmitted. The source can have many different forms such as an analog, biological, digital, or other kind of source.

We cover source coding in detail including the basic results of Kraft and McMillan leading to Shannon's First Theorem. This result gives the amazing connection between source coding and entropy, the subject of Chapter 10. The fundamental ideas of compression, which are crucial for such applications as data transfer or downloading from a source such as the Internet, are presented. In particular, arithmetic coding and the fundamental compression algorithms of Huffman and Lempel-Ziv are described. Redundancy is also discussed. The mathematical requirements in this chapter are not unreasonable even though we cover everything in full mathematical detail. One reason for doing this is that the remarkable connection between entropy and encoding must be shown to be believed!

New, Noteworthy The well known Entropy Lemma in Section 11.2 is at the heart of all source coding.

The proof that the Huffman algorithm is optimal is tricky. Various authors make an unwarranted assumption as follows. Given an optimal encoding and a source word of smallest probability, the corresponding code word will have maximal length. This code word will then have a "sibling" on the tree. The assumption, often made, is that the corresponding source word will have the smallest probability or the next smallest probability. In our proof here, we show how to avoid this assumption.

Chapter 20

Introduction to Linear Codes

Goals, Discussion We introduce the basics of linear codes and give several examples including the "perfect" Hamming and Golay codes. The McEleice cryptographic protocol is described as an application of the theory of linear codes.

New, Noteworthy We include a discussion of the "football pools" problem. Basic ideas from Shannon's fundamental theorem are revisited using Hamming codes over a binary symmetric channel. We discuss some of the fascinating history of the perfect ternary Golay code of length 11, which was in fact discovered independently and published in the Finnish soccer magazine Veikkaaja a year and a half before Golay published the same code.

20.1 Repetition Codes and Parity Checks

We want to revisit some of the general ideas of Chapter 19 and to amplify on them. Mathematical error correction is concerned with the errors which occur when information is transmitted from one place to another. For example, the binary message 010101 may be received as 011101 because of a problem in the transmission process. These problems are referred to as "noise" and may be due to electromagnetic radiation, thermal radiation (heat), cross talk, deterioration of storage devices such as hard disks, or even operator (human) error. The detection of errors is important because it sometimes allows for correction by transmitting the information a second time. The correction of the received message to the correct message is even more desirable, but usually harder to accomplish. Both of these objectives are realized by transmitting the information in a redundant form. For example, the binary message 010111 could be repeated three times by transmitting 010111010111010111. Then if only one error was made in the 18 digits transmitted, it would be easy to pick the correct message, since the corrrect message would occur twice and the wrong message would