Books

Contemporary Cryptology: The Science of Information Integrity, Gustavus J. Simmons, Editor, IEEE Press, Piscataway, New Jersey, 1992. 640 plus xv pp. (ISBN 0-87942-277-7).

Cryptology is the study of secret codes. But it is more than just hiding secrets, and it is applicable to far more than just military and diplomatic circles. This collection of survey articles provides a rich perspective of the uses of cryptography for: privacy, authentication, digital signatures, identification, authorization, certification, witnessing, liability, time of occurrence, voting, and many more.

This is a lively collection, written by authors prominent in the open research in cryptography, among them Whitfield Diffie, Gustavus Simmons, Ernest Brickell, James Massey, Andrew Odlyzko, and Judy Moore. The best of its chapters are instructive, informative, fascinating, and provide valuable reference work. The lesser chapters are sometimes written for a limited audience or are careless with notation. On the whole, it is an outstanding book.

The book grew out of an issue of the *Proceedings of the IEEE* that was devoted to cryptography (Volume 76, No. 5, May 1988). Most articles have been expanded and updated in the transition from the *Proceedings* to the present book; we find mention of research from 1990–1991 in factoring and differential cryptanalysis. A second printing corrected errata and updated some text, and a planned third printing will continue to improve.

Each chapter has its own complete bibliography, and there is an extensive index for the entire book.

Because the chapters are written by different authors, there is some overlap in the topics, but generally with different viewpoints. So in four different chapters we see RSA (Rivest-Shamir-Adelman) signatures in the context of cryptanalysis, protocols, applications, and the history of ideas.

M. E. Smid and D. K. Branstad give a chapter on the DES (Data Encryption Standard), emphasizing the history, the standardization process, and some applications. They also explore the future, the way in which the CCEP (Commercial COMSEC Endorsement Program) is intended to provide future standardized cryptographic algorithms. Technical detail is lacking here: DES is not fully described, either here or elsewhere in the book; and the various standardized "modes of operation" are presented without mention of the flaw in k-bit output feedback mode with $k < 64$ (see Chapter 10). But the perspective is valuable. Consider the closing quote: "But perhaps the most important contribution of the DES is that it has led us to other security considerations, beyond the algorithm itself, that must be made in order to have secure computer systems and networks."

A competing method of encryption is "stream ciphers," where messages are altered under the influence of a time-varying key. R. A. Rueppel describes some current implementations of stream ciphers, most of which involve shift registers, combined in various nonlinear fashions. We know only a few measures of strength of a stream cipher, two of them being "correlation" and "linear complexity"; these measures are applied to the various stream ciphers presented. Other schemes, more theoretical and more computationally expensive, are also presented.

A lively account of the early history of public key cryptography is given by W. Diffie. The emphasis here is on the evolution of ideas. Two ideas from

the 1950s—the challenge-response identification of aircraft, and the one-way functions protecting logon passwords—merged to form the idea of a trapdoor one-way function, and the potential applications of such a function were explored. Only later was a concrete function given that realized the idea. We are treated to a tour of the early schemes: exponential key exchange, trapdoor knapsacks, and the RSA number theoretic cryptographic system. We share the wonder in discovering the versatility of a two-key (public key) system: the same mechanism that allows digital signatures (only Alice can write this message, but anyone can read it) also allows secret communication (anyone can write a message, and only Alice can read it). We see how these schemes were related to the ideas of complexity in theoretical computer science of that day, and how they have influenced computer science research since then. The maturing field of public key cryptography has since found applications, commercial products, and a renewed interest in computational number theory.

J. Nechvatal follows with a more extensive survey on public key cryptography. He gives details of several schemes, as well as applications to authenticity, key management, secrecy, and certificates. Digital signatures are treated, along with the hash functions that make them practical. He discusses authentication protocols: by a series of certificates and hand-shaking messages, users can reliably identify one another and be sure of the source of messages. Thorny issues include implementation details, and the potential loss if a certifying authority is compromised, or if an ordinary user is compromised. An appendix contains a handy summary of the relevant mathematics.

One chapter deals with current progress in integer factorization (aimed at the RSA scheme) and in index calculation or "discrete logarithms" (finite fields) (useful for the Diffie-Hellman and El Gamal schemes). P. C. van Oorschot gives theoretical descriptions as well as estimated operation counts for relevant parameters, attempting to compare the security of N bits of RSA versus M bits of El Gamal. He also explores the strength of elliptic curve cryptographic systems, but inexplicably confines his remarks to those systems that can be reduced to logarithms in finite fields and solved there; the general elliptic curve system cannot be reduced to discrete logarithms, and appears to be much stronger than those treated here.

C. J. Mitchell, F. Piper, and P. Wild give a chapter on digital signatures. Analogous to an error-correcting code at the end of a transmitted message, which helps to detect accidental changes in messages due to transmission noise, a digital signature can help determine whether a message was intentionally tampered with. Between a mutually trusting sender and receiver, simple precautions suffice. But if the receiver might falsify messages, or the sender might deny having sent a message, more sophisticated techniques are needed. Many kinds of signatures are described here, some involving an arbiter on each message, some involving enormous computation loads, and some more practical. A necessary adjunct is a "hash function," which reduces a message to a smaller digest to which the digital signature is applied. Many hash functions are described here, and most of them broken.

My own favorite chapter is "Cryptanalysis: A Survey of Recent Results" (E. F. Brickell and A. M. Odlyzko). This is a lively survey of promising cryptographic schemes— knapsacks, variations on RSA, Ong-Schnorr-Shamir signature schemes— interlaced with the techniques used to break these schemes. Emphasis is given the Lovasz lattice basis reduction (for the knapsacks) and number theoretic techniques (for Ong-Schnorr-Shamir), versatile cryptanalytic tools. The powerful technique of differential cryptanalysis, made public quite recently, is mentioned only in passing.

In the companion chapter, "Protocol Failures in Cryptosystems" (J. H. Moore), we find that even a secure cryptographic scheme can be made insecure by an improper implementation. For example, if a public key system is used to encrypt one of a handful of possible messages, the opponent can encrypt the entire suite of messages and compare ciphertexts to discover which of the messages was sent. Another example given is when RSA is implemented with a small public exponent (say 3), and one message is sent to three different users, thus encrypted under three different RSA moduli; the opponent can combine the three ciphertexts and easily deduce the plaintext.

These two chapters, taken together, are instructive for the cryptanalyst (one who would break a

system) and the cryptographer (the designer of a cryptographic system) alike; by seeing how weaknesses in other schemes are exploited, one can hope to avoid such weaknesses in one's own schemes.

One exciting application of cryptography, still under development, is that of "smart cards." While an ordinary magnetic-stripe credit card carries information that can be read by a terminal (and copied by fraudulent terminals), a smart card also has an on-card processor. So the smart card can participate in zero-knowledge transactions, proving its authenticity without yielding its secrets. The chapter on this subject guides us through some political concerns in the standardization of smart cards; the commercial and political reasons for requiring integrity without confidentiality; the technological growth that allows ever-increasing processing power on a credit-card-sized device; and descriptions of various types of memory and their hardware costs.

Simmons himself contributes several chapters, perhaps the most fascinating of which is the last chapter on treaty verification. To verify compliance with the US-USSR treaty on nuclear testing, seismic detectors are buried in one country (the "host") and transmit data on underground tests to the other country (the "monitor"). Simmons gives us an inside view into the delicate negotiations and the cryptographic techniques for solving some problems of mutual distrust. Is the host substituting a false signal to mask the fact that it is continuing tests? Is the monitor really using the device to transmit other information than that allowed by the treaty? Who supplies the hardware? Can that person cheat?

Other chapters deal with shared secrets, zero-knowledge interactive protocols, and information authentication.

The book has several chapters of interest to the lay reader, and has enough mathematical content to satisfy people with a more professional interest. I highly recommend it.

Don Coppersmith
IBM Research Division
Yorktown Heights
New York

The Cleanroom Approach to Quality Software Development, Michael Dyer, John Wiley & Sons, Inc., New York, 1992. 198 pp. (ISBN 0-471-54823-5).

If you are looking for a way to produce better software, read this book. Mike Dyer may not answer all your questions but he does describe a revolutionary approach—the Cleanroom—and he will likely convince you to seriously consider using it. This book is pertinent to both managers and practitioners, and it should be read by anyone who is concerned about producing better software.

New departures are often shocking, and this one is no exception. People do not like to be told they are living in the past. Getting people's attention is the first problem; convincing them to try something new is the next. One approach is to write a convincing book that puts all the evidence in one place. This is such a book.

Harlan Mills first introduced his structured methods for software design over 20 years ago. By 1980, he, Mike Dyer, and others at IBM had developed these concepts into the full-fledged Cleanroom method. It has now been used by several groups, so preliminary data on its feasibility and effectiveness are available. The case so far is quite convincing.

The Cleanroom approach is a disciplined approach to producing high quality software. It is founded on the proven principle that doing the job right the first time is both faster and cheaper than building a poor product and fixing it later. While most of modern industry has learned this lesson, software is encumbered with an antiquated test-and-fix practice. This may seem a harsh judgment of many dedicated and hard-working software practitioners, but there is no question that software people almost universally rely on testing to find and fix the bugs in their products. That is why most software development organizations spend from 40 to 60 percent of their time on testing, and why the costs of later fixing defective products often exceed the development costs. It is not that everyone is unintelligent or stubborn, but that new methods are needed. The evidence given in this book suggests that the Cleanroom approach is, if not *the* new method, at least an important step in the right direction.

The principle behind the Cleanroom approach is that high-quality programs must start with high-quality designs. The method thus starts with a mathematically-based design approach, coupled with disciplined reviews to assure program correctness. Programmers have long known that no reasonable amount of testing can find all the problems in a program. It is simply uneconomic to run enough tests to assure that any but the most simple programs are defect free. Since testing is such a weak reed, Cleanroom removes it from the design process.

The design must be made correct without relying on the inadequate crutch of computer testing. When the programmer is convinced that the program is correct, it is submitted to a separate group for testing. Because humans are fallible, there are usually some latent errors; but these are generally trivial and can be rapidly fixed by the developer at the start of formal testing. From here on, the independent test group statistically examines the program to determine its quality. They follow a rigorous statistical procedure to ensure that the proper data are gathered and analyzed to determine when the program meets its quality criteria.

While this may sound time-consuming and expensive, Dyer shows that the Cleanroom approach is both more cost-effective and faster than the traditional test-and-fix method. What is more, the evidence also shows that Cleanroom produces higher quality products that cost less to maintain.

The book opens with a description of the Cleanroom method and its impact on software organizations. A COBOL project is described that produced 52 015 lines of code with a total of only 179 errors in all of its testing. At 3.4 errors per thousand lines of code, this is a remarkable 15-to-20-times better than industry practice. The resulting product had only 10 errors in customer use. This is 0.2 error/kloc, which is 10 to 50 times better than industry norms.

Following the introductory chapter, Dyer summarizes the lessons learned from Cleanroom experience and the problems of introducing Cleanroom into software organizations. He talks about getting started, the importance of training, and the key planning concerns. The following three chapters on how-to's briefly review the essentials of the Cleanroom method. While the material is easy to read, these chapters do not go quite far enough to serve as complete guides to the user. Either the referenced articles must be consulted or professional help will be needed.

I also found the chapter on software reliability a bit of a diversion. I would have preferred a more complete explanation of statistical testing. Since this is not covered in detail, the reader must consult an earlier article by Currit, Dyer, and Mills ("Certifying the Reliability of Software," *IEEE Transactions on Software Engineering* **SE-12**, No. 1, January 1986). Even with this reference, most practitioners will likely need help in mastering the intricacies of statistical testing.

Mike Dyer is well qualified to write this book. He has worked in software development for over 35 years and at IBM for 25 years. He worked with Harlan Mills in IBM's Federal Systems Division (FSD), both on the development of the Cleanroom method and in the program to introduce improved software engineering methods throughout the IBM FSD division. He has authored many articles, has lectured extensively, and is an experienced teacher.

The failings of this book are primarily its omissions. On the other hand, one of the book's great charms is its compactness. In a few hours you have the foundation to pursue Cleanroom in more detail if you so choose. In summary, the book's faults are minor; it is a *must read* for all software practitioners and managers.

Watts S. Humphrey
SEI Fellow
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA

Designing the User Interface: Strategies for Effective Human-Computer Interaction, Second Edition, Ben Shneiderman, Addison-Wesley Publishing Company, Reading, Massachusetts, 1992. 573 pp. (ISBN 0-201-57286-9).

The first edition of this book, published in 1987, was favorably reviewed in this journal that same year by L. Tetzlaff (*IBM Systems Journal* **26**, No. 2, 1987). Tetzlaff's review of the first edition still applies to the essential goals and substance of the

book. Shneiderman has written a useful introductory survey of user interface techniques, design guidelines, the research on which these are based, and methods of user-centered design and evaluation necessary for developing usable software interfaces. This edition remains a useful introduction to essential concepts and techniques in the broad multidisciplinary domain of human-computer interaction (or HCI). Chapters are well documented with references to original source materials, so readers interested in more depth have pointers to it.

The discipline of HCI has evolved in the last five years, and Shneiderman's second edition has been revised to keep pace with these changes. There are new chapters on user interface technologies that have matured since the first edition into research areas in their own right. These include computer-supported cooperative work, new visual information retrieval and exploration techniques, including discussion of hypermedia, and virtual reality, among others. Important familiar topics have been expanded or rewritten to take into account new developments. This edition devotes a complete chapter to discussion of graphical user interface windowing environments. Treatments of on-line help and menu design have been updated and expanded.

A desirable feature of the book is gaining familiarity with the author himself. Shneiderman is quite active and visible in the HCI field. He has been responsible for identifying useful concepts and research directions in user interface technology. For example, he coined the term "direct manipulation," and was one of the first to discuss its importance in making user interfaces more intuitive to computer users. He has carried out behavioral research in key technologies like hypertext and new, direct manipulation approaches to information retrieval. Shneiderman conveys his personal commitment to, and enthusiasm for making computer technology accessible to people. The book includes advice about how to break into the HCI discipline, and a survey of professional societies and information resources useful for newcomers. The author's interest in social issues is reflected in discussions of the needs of the disabled and elderly, and of the larger social and cultural impact of computer technology.

The coverage of interface technology, design guidelines, and background research is quite broad. Some topics, like "minimalism" in training and documentation, might have been discussed in more depth. And the book tends to emphasize interface techniques, rather than user-centered design and evaluation methods—which in my opinion, are critical to turning software technology into useful, usable tools for people. More discussion of these topics would have been welcome. However, once again, the reader interested in these methods will find references to more in-depth presentation. Overall, Shneiderman's book is a worthwhile update and expansion of an already useful introduction to the HCI discipline. It remains useful for professionals involved in any aspect of software development, where software usability, user interface quality, and even software innovation are objectives.

Robert L. Mack
IBM Research Division
Hawthorne
New York

Note—The books reviewed are those the Editor thinks might be of interest to our readers. The reviews express the opinions of the reviewers.