The Book Review Column[1]

by William Gasarch
Department of Computer Science
University of Maryland at College Park
College Park, MD, 20742
email: gasarch@cs.umd.edu

In this column we review the following books.

1. **The Cryptoclub: Using Mathematics to Make and Break Secret Codes** by Janet Beissinger and Vera Pless. Review by Abu Mohammad Omar Shehab Uddin Ayub. This is an intro to crypto for middle school children. And it could teach you a think or two!

2. **Cryptanalytic Attacks on RSA** by Song Y. Yan. Review by Leo Irakliotis. The book is a gentle introduction into applied number theory and in particular factorization in the context of cryptography and specifically the RSA system.

3. **Cryptanalysis of RSA and its variants** by Jason Hinek. Review by Antoine ROJAT. This is a serious book on RSA, its variants, and possible attacks on it. A tough but rewarding read.

4. **Understanding and Applying Cryptography and Data Security** by Adam J. Elbirt Review by Jérémy Barbay. This book describes cryptographical mechanisms and some of the mathematical concept required to understand them, from a very technical point of view, with a particular emphasis on the techniques of efficient implementation in hardware and software, as well as on the attacks that have been proposed in recent publications.

5. **Efficient Secure Two-Party Protocols: Techniques and Constructions** by Carmit Hazay and Yehuda Lindell. Review by Jonathan Katz. This is a monograph on efficient secure two-party protocols with an emphasis on work by the authors.

6. **Theory of Computation** by Dexter C. Kozen. Review by Daniel Apon. This is a textbook for a graduate complexity theory course. Each chapter is short and somewhat self-contained, making it nice to read parts of.

7. **Codes: an introduction to information communication and cryptography** by Normal L Biggs. Review by Robert J Low. The word *code* has several meanings. Overall you want to represent a string x by a string y. But why y? You might want a shorter string and use some sort of compression. You may want a string that does good error-correcting. You might want to make x not recoverable from y unless the receiver has a key. This book looks at all three notions.

8. **Finite Fields and Applications** by Gary L. Mullen and Carl Mummert Review by Jeffrey Shallit This is a short undergraduate textbook that gives applications of finite fields to combinatorics, coding theory, and cryptography.

9. **The Life and Times of the Central Limit Theorem** by William J. Adams. Review by Miklós Bóna. This book is about the history of the Central Limit Theorem. As such, many versions of it are gone through.

10. **Pearls of Discrete Mathematics** by Martin Erickson. Review by Robert Szarka. This book gives many short interesting... pearls (this is the best word I could find) from the field of Discrete math. What is a *pearl* ? One definition (emailed to me by the author) is that it is an interesting example that you might hear in a department coffee room.

11. **Design Theory** by C. C. Lindner and C. A. Rodger. Review by Dimitris Papamichail. Design theory is the part of combinatorics that deals with the existence and construction of block designs. Designs are collections of sets (or blocks) of objects (or symbols, points), with specified properties or features. In their general form, they can be defined by five parameters. They are used in design of experiments. How do construct them? Get the book and find out!

12. **An Introduction to the History of Algebra: Solving Equations from Mesopotamian Times to the Renaissance** by Jacques Sesiano. Review by William Gasarch. This book describes how people used to do Algebra. It is fascinating how clever and yet how primitive their methods were.

Books I want Reviewed

If you want a FREE copy of one of these books in exchange for a review, then email me at gasarchcs.umd.edu

Reviews need to be in LaTeX or LaTeX2e.

1. *In Pursuit of the Traveling Salesman: Mathematics at the Limits of Computation* By William Cook.

2. *Proof Analysis: A Contribution to Hilbert's Last Problem* by Negri and Von Plato.

3. *Graph Algorithms (2nd Edition)* Shimon Evan (edited by Guy Evan).

4. *Computability and Complexity Theory (2nd Edition)* by Homer and Selman.

5. *Computer Science: The Hardware, Software, and the Heart of it* Edited by Ed Blum and Alfred Aho.

6. *Physical-Layer Security: From Information Theory to Security Engineering* by Mattieu Block and Joas Barros.

7. *Bioinformatics for Biologists* Edited by Pavel Pevzner and Ron Shamir.

8. *Universal Semantic Communication* by Brendan Juba.

9. *The Block Cipher Companion* by Lars Knudsen and Matthew J.B. Robshaw.

10. *Theory of Conditional Games* by Wynn Stirling.

11. *Security and Game Theory: Algorithms deployed systems, lessons learned*

Review of[2] of
The Cryptoclub: Using Mathematics to Make and Break Secret Codes
by Janet Beissinger and Vera Pless
A K Peters, 2006
200 pages, Softcover

Review by
Abu Mohammad Omar Shehab Uddin Ayub (shehab1@umbc.edu)
Department of Computer Science and Electrical Engineering
University of Maryland, Baltimore County

1 Introduction

Cryptography is touching everyones life these days in both visible and invisible ways. We give passwords to log in FaceBook accounts, hit the key pad to enter secured houses, input credit card numbers while buying on Amazon. We think twice whenever the operating system warns us that the software we are about to install is not from a known or secured source. The web browser warns us if we are about to visit an insecure website. Kids around us see what we are doing and always ask what was that we input so fast and do not want to show to anyone (passwords). A kid who never tried to open a lock without a key is, even if highly welcome, very rare. Wishing to unveil the secret information is human. The Cryptoclub is a great journey about this nature through the classic and recent topics of cryptography palpable to the middle school children. The book is about how to obscure our precious information when we do not wish to share and how to be apprehended by someone we choose to share with.

2 Summary

The book is divided in seven units each containing several chapters. Each chapter starts with a beautiful illustration. A story plot spans the whole book throughout the chapters. Abby, Evie, Jenny, Daniel, Tim, Peter, Lilah, Jesse are the middle school kids divided into two friendly rival groups, boys vs girls. Each group is always trying to beat the other by using a new cipher trick or breaking other's tricks. The audience will easily find themselves matching with the characters as they proceed. The problems are not bunched up at the end of the chapter. The authors did a nice job by associating the problems of a chapter right with its corresponding sections. Each chapter introduces one or more tools to cipher messages. There are hands on tips in every chapter which helps the audience to make the tool at home. Each section gives hint about which problems should be approached by then. At the end of every chapter, there is a section containing interesting historical facts in a story telling way. To some readers, this might be the best part of the reading. At the end of the book, a standard-sized sample cipher wheel is added that can be ripped off and used right away. Here is a summary of all the chapters.

2.1 Unit 1: Introduction to Cryptography

2.1.1 Chapter 1

Abby tries to send a secret message to her friend Evie. Unfortunately it was intercepted by the teacher. Everyone in the class knew about it. Abby felt humiliated. So, there was a situation where messages needed to be sent so that if they are intercepted they will seem garbled. At this point the book introduces Cryptography. It starts with the Caesar cipher. After a number of examples, it finally presents the cipher wheel which can be used as a handy tool to generate Caesar ciphers.

2.1.2 Chapter 2

Jenny already knew a bit about ciphering. Now she wants to make it perfect. She was introduced to the Cipher wheel. Eventually she realized that she doesnt need to keep the wheel always with her. Simple arithmetic knowledge is enough to cipher her message. At some point her arithmetic overflew beyond the standard index of the English alphabet. Then she discovered that cycling back the numbers, when necessary, gives a nicely behaving ciphered message. So, she learned how to use arithmetic to do the same thing a cipher wheel does.

2.1.3 Chapter 3

Boys in the class start sending ciphered messages following the girls. The girls wanted to break them. The main challenge was to find out the key which was used to cipher the message. The boys were confident but the girls were able to break it. It turns out that breaking is more difficult than ciphering. To break the code they focused on finding the most frequent letter the boys have possibly used. With some trial and error, the trick worked for them. The boys were surprised. This chapter is about breaking the Caesar cipher by finding the key. The key is determined by assuming and trying the possibilities for the most frequently used letter.

2.2 Unit 2: Substitution Ciphers

2.2.1 Chapter 4

As the Caesar cipher or the shift pattern cipher was common in the school, the messages were being decoded by unwanted persons. To share about an upcoming ski trip with his friends, Dan wanted to use a stronger cipher. He replaces the letters of his message with random unique letters from the alphabet. It looked good but too clumsy to manage. He has to send the whole substitution table to every person he wants to share the message with. So, the chapter introduces Keyword cipher. A random chunk of the text is replaced by a keyword. Then the rest is replaced by the remaining letters in regular order. So, the message looked scrambled enough with no obvious shifting pattern. This time the cipher was stronger as it needed two pieces of secret information to decipher, the keyword and the key letter.

2.2.2 Chapter 5

Now its the girls' turn to break the boys' message. They tried with Caesar and other substitution ciphers and failed. It looked harder without any obvious pattern. But Jenny didnt let her girls down. She observed which letters occurred most often in the coded message. Then she collected

information on high occurring letters in English. By comparing this information, she was able to find out the hidden message. This chapter introduces the concept of relative letter frequency and how to collect sample data to determine relative frequency of different letters. It also contains a standard observation of relative frequency of letters in English language.

2.2.3 Chapter 6

After knowing about relative frequency the children wants to try code breaking with the new skill. They were surprised to find out that the relative frequency table from another source is amazingly similar to the table they worked out on the data collected by themselves. Peter requested Jenny to show how she broke the message in the previous chapter using the frequency table. She demonstrated how she matched the letter with highest relative frequency with the most frequently occurring letter in the message, which didnt always work. Then with some trial and error facilitated with smart guesses she was eventually able to determine the meaning of the message. She used the knowledge about common word construction patterns of English in a very tricky way.

2.3 Unit 3: Vigenére Ciphers

2.3.1 Chapter 7

The girls discover a coded message from Jenny's grandpa's old things. They discussed it in the next meeting of the newly formed Cryptoclub in the school. A new member of the club, Jesse, pointed out that not all the ciphers are breakable using relative frequency. He told about a new system called Vigenére cipher. In this chapter the system is demonstrated using the good old cipher wheel. Finally the Vigenére square is introduced, which made the coding and decoding faster.

2.3.2 Chapter 8

In this chapter, the club members decided to do some exercises with the newly learned Vigenére cipher. Boys and girls made up two separate groups and wrote a cipher message for the other group to decode. They thought that if the length of the keyword is known as a hint, life would be easier. Now they were able to determine which cipher wheel was used to encrypt each letter. Although it was lengthy and tedious, each team was able to decipher the message of the other team. They had a good exercise with the Vigenére cipher.

2.3.3 Chapter 9

In this chapter the kids felt that before starting with the grandpa's message they should review some math. They reviewed the concepts of factor, multiple, prime and composite numbers, divisibility, prime factorization, factor tree, exponent, greatest common factor, etc. This is the fundamental math running behind any cryptography system.

2.3.4 Chapter 10

Now the kids were feeling more confident about breaking ciphers. They learned how to factor a coded message into its components. They discovered that in Vigenere cipher, string patterns are repeated in patterned distances. If a number of strings are identified as repeated, breaking the

message becomes a whole lot easier. They tried to find the repeated string in grandpa's message. From that they had a good guess of the length of the keyword. They used the techniques used in previous chapters and finally were able to break it. It was a big moment. But when they went to share it with Jenny's mom, a whole different story was waiting for them.

2.4 Unit 4: Modular (Clock) Arithmetic

2.4.1 Chapter 11

Curious Tim wanted to know why we have identities for arithmetic operations. His teacher failed to satisfy him with her answer. He discovered that 10 + 10 is not always 20. On a wall clock it is 8. He called this newly discovered arithmetic as clock arithmetic. Then the chapter introduces another version of clock arithmetic with a twenty four hour clock. This arithmetic also has a different result for arithmetic operations. When their teacher found the discoveries she pointed out that there is already a branch in arithmetic called modular arithmetic. The kids found the modular arithmetic operations amazingly fast.

2.4.2 Chapter 12

Kids played a lot with modular arithmetic. They learned how to reduce numbers by mod operation. Then they used this idea to implement the times-11 cipher. They discovered that with modular arithmetic they can do large multiplications pretty fast. The teacher also showed that they can use it to predicted which day of the week a future date will be.

2.5 Unit 5: Multiplicative and Affine Ciphers

2.5.1 Chapter 13

In the previous chapter the kids used multiplication by 11 to generate a table a cipher. They decided to play more with multiplications. They generated tables by multiplying with different numbers. There were good and bad tables. The bad tables generate ambiguous coding but the good tables don't. Numbers relatively prime to 26 turned out to be good cipher keys to multiply with. It was a significant discovery. The modular arithmetic learned in previous chapters was useful to explain why the numbers which were not relatively prime were not good cipher keys. Eventually, the kids discovered that a good cipher key for the English alphabet can be a bad key for the Russian alphabet because of different number of symbols in the two alphabets.

2.5.2 Chapter 14

Ciphering messages by multiplication became popular in the school. As it was used more and more, people needed to decipher them. Abby remembered the way they used addition and subtraction to encrypt and decrypt. She thought division may help to decode multiplicative ciphers. It was partially successful. Some of the letter indices were not perfectly divisible. Then the chapter introduces the ideas of multiplicative inverse and reciprocal in terms of modular arithmetic. Determining modular inverse needs the knowledge of multiplicative inverse. They were successful to decrypt messages coded with multiplicative ciphers.

2.5.3 Chapter 15

At one point Dan and Tim thought they should change their cipher key frequently so that other students will not be able to read their messages even if they remember the keys used previously. It turned out that the English alphabet does not have enough symbols even for two months so that they can choose a unique good cipher key every day. They decided to do both additive and multiplicative cipher together. This system is called the affine cipher. It has a pair of cipher keys one for multiplication and the other for addition. Decryption was pretty straightforward. They need to reverse the arithmetic operations to decode the message using the same keys. The boys faced a challenge when the girls posted a coded message. They realized that it used an affine cipher but they didnt have the key. So, the chapter introduced modular algebra which helped the kids to determine the keys by solving modular equations.

2.6 Unit 6: Math for Modern Cryptography

2.6.1 Chapter 16

By now the kids became really good in handling ciphers but they were just centuries old techniques. They wanted to learn something modern. Tim said he read an article about RSA cryptography which used very large prime numbers. So this chapter discussed different properties of prime numbers. They learned how to check primality using arithmetic and the Sieve of Eratosthenes. Then they tried to determine *all the prime numbers* using those techniques. Lilah disappointed him by informing that the Greeks already knew 2000 years ago that there are infinitely many prime numbers. They also learned the formulas for twin, Mersenne and Sophie Germaine primes. Finally they were very surprised to know that there are still lots of discoveries taking place in mathematics and lots of old questions are still unanswered.

2.6.2 Chapter 17

In their pursuit to learn the math required for modern cryptography, the kids learned how to deal with numbers which are raised to some power in modular arithmetics. It was discovered that even the super large numbers are easily manageable in modular arithmetic.

2.7 Unit 7: Public Key Cryptography

2.7.1 Chapter 18

So far the students have been sending keys to their allies secretly. But unwanted people can intercept the key as they intercept the coded message. It will no longer remain secret. Managing secret key sharing became a big challenge. So, the chapter introduced public key system. With a pair of public and private keys, they were able to share both the message and the public key with their friends. As the private key of a friend was still secret, no one other than she was able to read the message encoded with the public key. They learned how to use prime numbers and their products as keys to encrypt and decrypt messages in RSA. It was amazing to know that people even use 200 digits long numbers and their products as keys in real life.

2.7.2 Chapter 19

To get a better understanding about how RSA works, this chapter revisits the ideas of modular arithmetic. The kids learned several new properties of modular inverse. Now they feel more confident to encode their own RSA messages.

2.7.3 Chapter 20

The kids built a directory of public keys of all students. So, to send a coded message to Tim, Dan encrypted the message with Tims public key and posted the message in public. This message can only be decoded with Tims private key , which is unknown to everyone else. He decoded and replied to the message using Dans public key this time. Eventually everyone in the class was able to send secured messages to everyone else very easily.

3 Opinion

This book is perfect to teach cryptography to students from grade six and beyond. Even students from early college years may also use this book as a quick refresher course. This is a very well written and interactive book with lots of illustrations and tips. Those who are interested about history will also find it interesting as every chapter concludes with a story about the history of a particular cipher. The worked out examples are correct. If you find this book interesting or the review useful let me know at TIFIBC2BUVNCDEPUFEV!

Review of[3] of
Cryptanalytic Attacks on RSA
by Song Y. Yan
Springer 2008
254 pages, HARDCOVER

Review by Leo Irakliotis
leo@nova.edu
Graduate School of Computer and Information Sciences
Nova Southeastern University, Fort Lauderdale, Fla.

1 Introduction

The book is a gentle introduction into applied number theory and in particular factorization in the context of cryptography and specifically the RSA system. Since its introduction in 1997, RSA's vulnerability has been studied and well understood. When attacks again weakly-keyed RSA systems prove partially successful, stronger (longer) keys are introduced and the chase continues. Given the complexity of the factorization problem, today we feel safe with 4096-bit or longer keys. Nonetheless, efforts to break the RSA persist and the topic of attacks on the RSA system merits a good review for those interested in studying the matter further. This is where Yan's book comes in handy.

2 Summary

The book begins with a good review of computability and number theory. The brief introductions to the euclidean algorithm, the Chinese remainder theory, and complexity classes are very helpful for first-time readers and useful as a refresher to more experienced readers. Public-key cryptography and the RSA system are introduced next. These introductions cover the first two chapters of the book.

The next nine chapters provide a taxonomic review of various attack methods that have been developed since the introduction of the RSA system. Yan covers quite a few attacks, grouped according to their methodology. The grouping follows a generally increasing level of sophistication: from integer factorization and discrete logarithmic attacks (in chapters 3 and 4), to quantum computing attacks, to private and public exponent attacks, and finally side-channel attacks. A second taxonomy exists whereby attacks are grouped in direct methods (find the RSA modulus) and indirect methods (lower exponents etc).

The last and very brief 10th chapter deals with the future of RSA and its further strengthening should quantum methods become effective. The chapter covers elliptic curve, code-based, and lattice-based systems as well as quantum cryptography.

Chapter 1 begins with an introduction to Turing machines. Complexity classes are discussed next and are contextualized with various number theoretic problems that are intractable or efficient

[3]

to compute. The discussion of course includes the integer factorization problem as a prelude to the success of the RSA system.

In chapter 2 Yan introduces the RSA system and useful variants including elliptic curve RSA and ElGamal. The chapter follows an interesting pattern where each asymmetric system discussed is analyzed in terms of its cryptographic, cryptanalytic, and cryptologic characteristics.

Chapter 3 describes various methods for factoring the RSA modulus. The methods are grouped according to their running time dependence on the size of the modulus or the size of the modulus' factor. Number field and quadratic sieve techniques are discussed in detail. The author describes how multiple-polynomial sieve technique was successful used in factoring RSA-129. The chapter concludes with a brief discussion on strengthening the RSA modulus against factoring attacks.

In chapter 4 the author discusses the discrete logarithmic problem and how related algorithms can be used to attack RSA. Techniques based on this problem include the baby/giant-step attack (a brute-force variant that creates a sparse list of values where the possible solution may lie within); the Silver-Pohlig-Hellman attack, and calculus attacks. Again the chapter concludes with brief remarks about strengthening an RSA key against logarithmic attacks.

Chapter 5 discusses quantum computing attacks. Considering the introductory approach the author takes in the previous chapters, readers might find chapter a bit more challenging as the author offers no introduction to quantum computing basics. Two quantum order finding techniques are used to illustrate the vanity of RSA encryption if or when quantum computers become a commodity.

In chapter 6 Yan describes elementary attacks on RSA, such as guessing the values of plaintexts. Such attacks exploit vulnerabilities arising from improper use of asymmetric encryption.

Chapters 7 and 8 discuss attacks that focus on RSA's public and private exponents respectively. These attacks rely on poor choices of exponents, to some extent and, as such, can be viewed as elementary. Coppersmith's theorem is reviewed as a trick that enables attacks on short public exponents (why would users opt for $e = 3$ these days, thus opening themselves up to short e attacks is beyond my grasp). Nonetheless, Yan's review on public exponent attacks leaves readers with a few good tips about avoiding such attacks. Similarly poor choices for the length of the private exponent d can increase the vulnerability of the RSA-system.

In chapter 9 Yan discusses vulnerabilities that may be exploited at the system level. Kocher's time technique, for example, that finds the private exponent d using data about the system's computational performance is a typical side-channel attack. As such attacks require some proximity or physical access to the target system they are not cryptanalytic per se.

The last chapter of the book provides a brief discussion about techniques that can strengthen classical (as opposed to quantum) cryptography using asymmetric keys as well as a discussion about the capabilities of quantum cryptography.

3 Opinion

The book offers a sufficient introduction to direct and indirect techniques for attacking RSA-encrypted systems. It contains several examples but the lack of problems may preclude its use as a textbook in a course. I would not hesitate to cite it as recommended reading for a course in information security however.

What I found lacking in Yan's book is code snippets from actual implementations of RSA attacks, computer performance metrics and statistic from such realistic attacks, and a discussion

about optimizing attacks through distributed or parallel processing (other than quantum computing).

Overall it is a pleasant book to read and a useful supplement for a course in information security.

Review of[4] of
Cryptanalysis of RSA and its variants
by Jason Hinek
CRC Press, 2009
268 pages, HARDCOVER

Review by
Antoine ROJAT (antoine.rojat@prism.uvsq.fr)

1 Introduction

In cryptography, the asymmetric encryption paradigm allows two parties to communicate without a pre-shared secret. Only authenticated public parameters are required in order to ensure the security of their communication. The idea is to use a trap-door function : a function that is easy to compute but impossible to inverse without the so called trap-door. The most well known instantiation of this paradigm has been proposed in 1978 by Ron Rivest, Adi Shamir and Leonard Adleman:
the RSA cryptosystem.
This cryptosystem is nowadays the most widely used (most of your online banking traffic rely on RSA for example). That is why since 1978, the study of the RSA cryptosystem is a very active research area. Every cryptanalyst is trying to find attacks against this cryptosystem. An attack is the exploit of a flaw in a cryptosystem. Using a flaw an attacker might be able to recover the secret parameters or he might just be able to recover a plaintext from a ciphertext.
This book traces the different attacks that exist against the RSA textbook version since breaking this version of the cryptosystem will allow someone to break any others.

2 Summary

The book is divided into three parts. The first one describes the RSA cryptosystem and gives some mathematical background which is necessary for describing the different attacks. The second parts focuses on different kind of attacks against the classical RSA and the last one presents the variants of RSA and some attacks on those variants.

The first part is divided into two chapters. The first one gives an overview of the RSA cryptosystem and some of its variants. It also include security and efficiency consideration. The second one briefly recalls some mathematical results necessary for understanding the attacks. In this chapter, one can find results about simple modular arithmetic (Chinese remainder theorem, solving modular linear equations), continued fractions, lattices (LLL, shortest vector problem and a long analysis of Coppersmith's methods. While presenting the mathematical results, the author also explains how they are going to be used in order to build an attack. For example, when describing the Coppersmith's methods, the author explains that in order to find small roots of a multivariate polynomials that does not satisfy a specific condition, one can build a polynomial that have the same roots and that satisfies the condition.

The second part of the book is divided into five chapters: The first chapter describes attacks that use a concrete instantiation of the cryptosystem and misbehave with it The second chapter presents attacks that exploit the fact that a cryptosystem has a small public exponent. It is important to consider this setting since in practice, it is generally assumed that the person encrypting data has a very small computational power. None of the attacks described in this chapter allow a total break of the cryptosystem, but assuming some knowledge of a plaintexts, and small public exponents, it is possible to recover a plaintext. The third chapter regroup the attacks that exploit the fact that the private exponent is small. Most of the results presented within this chapter allow an attacker to recover the private exponent and thus to break the cryptosystem. The fourth chapter focuses on leakage : what if the attacker has access to some bits of one of the private parameters ? How does this affect the security of the cryptosystem ? The fifth chapter is exploring the small private exponent flaw combined with others such as "common modulus" or "common private exponent". In this chapter, the author assumes that an attacker knows that there is a small exponent and furthermore, he also has a small hint. The question is : how does this hint affect the security of the cryptosystem ?

The last part presents some of the possible variants of the RSA cryptosystem. Every time the author presents a variant, he reexamined the different kind of flaws and how the variant is affected. This part describes five variants of RSA. The first variant is the CRT RSA : during the decryption the Chinese remainder theorem is used. The second variant is the Multi-prime RSA : instead of using a moduli of two primes, one can use a moduli of k primes for $k \geq 2$. The third variant is Takagi's scheme ; the moduli is slightly different : instead of being the product of two primes, it is of the following form : $N = p^{b-1}q$ where p, q are primes and $b \geq 3$. The fourth variant is a classical RSA where the primes p and q have a specific form which allows them to resist to Wiener's continued fraction attack event though they have a large common prime. The last variant is the dual RSA which consists in two RSA instance sharing the same private and public exponents. Along with the flaws introduced by the variants and the different attacks presentations, the advantages of the different variants compared to the classical RSA are fully analyzed.

3 Opinion

I enjoyed reading the book because because the author is always caring about the different references that he used to write the text. This allows the reader to go further in his understanding of what is presented. I also appreciate the fact that the attacks presented in the book are first describe at a upper level before examining the details. I have to underline the fact that the necessary mathematical notions presented in the book are a bit fuzzy and some results might need more details. Furthermore there are some typos as an example on page 5, one can read: "messages that are relatively prime to the modulus (i.e.,$gcd(m, N) > 1$) should be avoided" ; obviously "not" is missing in the sentence. As the author followed the same structure in all the chapters, it is always easy to find a specific information in the book which is a very useful feature. Even though some stuff might need more attention from the author, I was pleased by the way the author presented all the different attacks and variant of RSA.

I would re command this book for people who would like to know more about the RSA and more precisely about the difficulties of attacking this cryptosystem with mathematical techniques. The

stuff presented in the book might seem quite difficult but I would still recommend this book for students wanting to improve their knowledge of cryptanalysis techniques since they can be reusable against other cryptosystem. For teachers, I would say that this book can be used as a base for building a complete course on RSA's cryptanalysis.

Review of
Understanding and Applying Cryptography and Data Security
by Adam J. Elbirt
CRC Press, 2011
637 pages, HARDCOVER

Review by
Jérémy Barbay, jbarbay@dcc.uchile.cl

Departamento de Ciencias de la Computación
Universidad de Chile
Avenida Blanco Encalada 2120, Tercer Piso,
Santiago, Chile C.P. 8370459
Tel: 56-2-978-4983, Fax: 56-2-689-5531

1 Introduction

This book describes cryptographical mechanisms and some of the mathematical concepts required to understand them, from a technical point of view, with a particular emphasis on the techniques of efficient implementation in hardware and software, as well as on the attacks that have been proposed in recent publications.

2 Summary

1. Introduction

   The author motivates the study of cryptography by covering shortly the past and current history of Cryptography and Data Security, and motivates the writing of this particular book by pointing out that the existing literature is mostly written from the point of view of mathematics, as opposed to engineering and computer science.

2. Symmetric-Key Cryptography

   In this chapter, the author introduces the basic definitions of cryptography (not only symmetric), and the basic mathematical concepts of modulo, greater common divisor, and ring, all to be used in the design of symmetric-key cryptography protocols (other mathematical concepts such as groups are defined later in the book).

3. Symmetric-Key Cryptography: Substitution Ciphers

   In this chapter, the author describes some basic principles of Cryptography, such as the Kerckhoffs' Principle, a classification of the type of attacks on cryptographic algorithms, and describes in detail toy examples such as Affine and Shift ciphers.

4. Symmetric-Key Cryptography: Stream Ciphers

After shortly motivating the need for cryptography on streams, the author gives some mathematical background on the generation of random numbers.

The author uses the presentation of the one-time pad protocol to introduce the notion of Unconditionally Secure, mentioning that "it is nearly impossible to state that something is proven to be secure" and that "The opinion of an algorithms or cryptosystem's strength of security is based on the scrutiny applied by cryptanalyst over time."

Arguing about the prohibitive cost of time pad protocols, the author describes several (imperfect) alternatives such as Key Stream Generators and Linear Feedback Shift Registers, and several attacks which can be performed on them.

5. Symmetric-Key Cryptography: Block Ciphers

   The author describes the notion of confusion and diffusion (introduced by Shannon in 1949) of a crypto-system, and in particular the avalanche system which provides diffusion.

   He then proceeds to describe the various components and modes of operation of the Data Encryption Standard (DES), technical details about its implementation, and savorous details about past attacks against DES, among other things the original suspicions of built-in backdoors, later proved to be wrong.

   The author then proceeds to describe how the Advanced Encryption Standard (AES) was designed, along with some more mathematical background about Galois Fields, and the technical details of its implementation and potential attacks against it.

6. Public-Key Cryptography

   After shortly motivating Public-Key Cryptography by criticizing the cost of key distribution in Symmetric-Key Cryptosystems, the author introduces the basic concept of Public-Key Cryptosystem and some mathematical concepts required in the following chapters on public-key cryptography (mainly for RSA), such as one-way functions, Euclidean algorithms, Euler's phi function and theorem, Fermat's little theorem.

7. Public Key Cryptography: RSA

   The whole chapter is dedicated to the description of the technical implementation of RSA, and of the evolution of the techniques and time required to attack RSA over time, which motivates the final recommendation "that 2048-bit moduli be used with the RSA algorithm to ensure long-term security."

8. Public-Key Cryptography: Discrete Logarithms

   After introducing the mathematical notions of Cyclic Groups and the Discrete Logarithm Problem, the author describes the Diffie-Hellman Key Agreement Protocol and the technical details relative to its implementation, and to techniques of attack on it, the "Baby-Step Giant-Step algorithm" which proceeds by finding the pair of integers (s, t) that requires a large amount of space ($1.5\sqrt{n}\lg n$ bits), and the "Index Calculus method", "by far the most powerful attack against Discrete Logarithm Cryptosystems".

9. Public-Key Cryptography: Elliptic Curves

After introducing the basic mathematics definitions of Elliptic Curves, and motivating their use by mentioning that "such cryptosystems require significantly smaller operands to achieve equivalent security", the author proceeds to describe the technical details relevant to the implementation of the "Diffie-Hellman Key Agreement Protocol", and the potential attacks against it.

10. Cryptographic Components

    In this chapter, the author describes in detail the concept and various systems for Digital Signatures (first proposed by Diffie and Hellman in 1976). Mentioning that a basic implementation has a prohibitive cost as it requires to break each message in smaller block, and to sign each separately, the author describes how this can be solved by combining cryptographic techniques with hashing techniques, which must be chosen wisely so that not to compromise the security of the digital signature system they complement. Distinct from Digital Signatures, the author finally defines Message Authentication codes (MAC), which are used to validate both the integrity and the authenticity of a message.

11. Cryptographic Protocols

    In this final chapter, the author describes the challenges posed by the combination of security mechanisms in order to provide a network with a set of security services. He defines a variety of attacks such as Interruption of data transmission, Snooping of Transmitted data, Modification and forwarding of data to a destination, and Fabrication and transmission of data, and gives again the list of services which must be provided (already given in the introduction): Confidentiality, Data Integrity, Message Authentication, Non-Repudiation, Entity Authentication and Access Control (the two latest separately, without any obvious reason?), and how the techniques seen in the previous chapters were combined in order to produce systems such as Kerberos, Pretty Good Privacy, Secure Sockets Layer, and Internet Protocol Security.

3 Comments

The book content is sometime a bit too informal: the definition of Cryptographic Protocol is a bit vague ("a sequence of steps that must be performed in a specified order to achieve a particular task": this could be an algorithm); the notion of security is vague, arguing that a function is hard to compute by showing the several pages of calculations "required" to compute it; several results are proved by giving an example (which is not a proof).

4 Conclusion

This book is a good technical reference for someone who wishes to implement (or install) a security system based on existing techniques, or even someone who wishes to learn how to attack existing security systems. It is a good complementary book for a technical course aimed at students who already know about the theory of cryptography but need to learn the technical details in order to implement existing protocols in new hardware.

Review of[5]
Efficient Secure Two-Party Protocols:
Techniques and Constructions
by Carmit Hazay and Yehuda Lindell
Springer, 2010

Reviewed by Jonathan Katz
Dept. of Computer Science, University of Maryland

1 Introduction

Textbooks and monographs (where a monograph is distinguished from a textbook by a monograph's focus on a relatively specialized topic) can play a significant role in defining a field. Books that are widely used in graduate courses can shape students' (and professors'!) perspectives on the subject; a text that becomes a 'bible' for graduate students can influence the direction a field moves. Textbooks and monographs can even have influence outside their own community, if they provide a treatment that is accessible to a broad audience from a wide range of backgrounds.

This potential impact of monographs (and, to a lesser extent, textbooks) appears underappreciated in our community. More to the point, there seem to be fewer texts available than there "should" be — observe how often graduate computer science courses are taught without any assigned textbook, or how difficult it can be to generate a reading list for a beginning graduate student — and certainly there are fewer TCS-focused monographs than there are in disciplines such as chemistry or mathematics, to take two examples. While there may be legitimate reasons that partially account for this, the result is to the overall detriment of our field.

This situation has, thankfully, begun to change. Several excellent textbooks have become available in the past few years, and we have also witnessed the publication of many monographs that have become quite popular within their own niche areas. The book under review, constituting a detailed treatment of efficient secure two-party computation suitable for a graduate seminar or self-study, continues this positive trend. (Full disclosure: I have co-authored papers with both authors, and have written a textbook with one of them.)

2 Summary of the Book

The focus of this book is efficient protocols for secure two-party computation. This area of research has experienced a surge of interest lately, both within the cryptography and security research communities as well as in terms of government funding (at least in the US and EU) for work on the problem. Secure computation itself has a long history dating back to the early 1980s. Roughly, a protocol for secure computation of a function f allows two parties, holding inputs x and y respectively, to jointly compute $f(x, y)$ while ensuring several security properties, chiefly *privacy* (nothing is revealed to either party about the other party's input beyond what is revealed by $f(x, y)$) and *correctness* (neither party can cause the other party to output an incorrect result). Seminal results of Yao and Goldreich, Micali, and Wigderson show that *any* (polynomial-time) function f can be computed securely, based on standard cryptographic assumptions, in either the semi-honest

or malicious settings and for any number of corrupted parties. (In the semi-honest setting, parties are assumed to follow the protocol but may then try to learn additional information about the other parties' inputs. In the malicious setting, parties can behave arbitrarily.) These results are among the most important in the field of cryptography, and contain beautiful ideas that every theoretical computer scientist should be aware of.

Initial results on secure computation focused primarily on feasibility. More recently researchers have developed and implemented highly efficient protocols for certain tasks. Roughly, research has progressed in three (overlapping) directions:

- Exploring (reasonable) weakenings of the security definitions in the hopes that these relaxed definitions will enable more efficient protocols.
- Optimizing protocols for "generic" secure computation that can be applied to arbitrary functions represented as boolean circuits.
- Developing efficient "special-purpose" protocols for particular functions of interest. These protocols exploit specific properties of the functions under consideration, and so do not (necessarily) rely on a boolean-circuit representation.

The present book describes results of the authors in each of the above areas. Following an overview of the book in Chapter 1, the book presents formal definitions of security for secure computation in Chapter 2. In addition to defining the "standard" notions of semi-honest and malicious security, the book also includes various intermediate notions including nonsimulation-based definitions of security and a newer definition proposed by Aumann and Lindell called *covert security*. Roughly, covert security does not *prevent* a malicious adversary from "cheating" and potentially violating various security guarantees; rather, it merely guarantees that any such cheating will be *detected* with high probability (at which point the second party can seek legal recourse). It has been argued that in many real-world settings, this threat of being caught will be sufficient to deter malicious behavior in the first place.

Chapters 3–5 describe "generic" protocols for secure computation of arbitrary functions f based on a boolean circuit computing f. Chapter 3 contains a full description of Yao's protocol for semi-honest adversaries, along with a detailed proof of security. (Amazingly, Yao's original paper includes neither the details of the protocol nor a security proof!) Chapters 4 and 5 discuss how to extend Yao's protocol so as to achieve malicious security and covert security, respectively.

The remainder of the book focuses on efficient protocols for specific functions. Thus includes several protocols (achieving different notions of security) for *oblivious transfer*, a fundamental primitive that is used as a building block of the generic protocols for secure computation mentioned above; protocols for oblivious pseudorandom function evaluation; for finding the median (or, more generally, the kth-ranked element) of two lists held by the parties; for keyword search; and for pattern matching.

If there is a fault with the book, it is that it focuses largely (though not exclusively) on results of the authors. These results do tell a coherent story; still, there were several results by other researchers that, in my mind, should have been included in a survey text of this sort. One prominent example is the work on *oblivious transfer extension*, which allows a large number of oblivious transfers to be carried out at essentially the cost of only k oblivious transfers (for k a statistical security parameter) and is the method of choice in practice when oblivious transfer is implemented. The authors also do not cover any results in the *random-oracle model*. Although there may be

valid theoretical justification for omitting such results, protocols in the random-oracle model are generally more efficient and would again be preferred in practice. It would have been helpful if the authors had at least included pointers to this other literature at the end of every chapter.

3 Recommendation

Overall, the book is a pleasure to read, containing sufficient motivation, intuition, and informal discussion as well as detailed proofs of security. The book contains a superb treatment of both general secure two-party computation as well as several efficient protocols in this setting. The first three chapters of the book would serve as an accessible introduction to secure two-party computation for the interested graduate student; the rest of the book is an excellent starting point for the more specialized literature in the field. The book could also serve very nicely as a text for a graduate seminar in this area, or could even be used as a supplementary book at the end of a graduate "Introduction to Cryptography" class. (I am planning to use it this way later this semester.) It belongs on the shelf of every researcher interested in this area.

Review of[6]
Theory of Computation
by Dexter C. Kozen
Springer, 2006
418 pages, Hardcover, $71.86 (Amazon)

Review by
Daniel Apon
dapon@cs.umd.edu

1 Introduction

Theory of Computation is designed to serve two purposes: (i) provide a student's first, rigorous survey of the foundations of computing, and (ii) give a taste of an assortment of advanced topics, to provide an avenue for further study. Its most immediate, striking aspect is a unique organization. Rather than using a chapter format, the book is divided into a series of "lectures." Each lecture is between 4 and 7 pages long and is designed to be a self-contained, readable unit on some topic. I found this approach extremely appealing.

The content of the book primarily focuses on computational complexity theory, though it briefly covers other relevant topics – for instance, there are a few lectures on algorithms for factoring and another on the basic bounds and inequalities used in probabilistic algorithm analysis. It's useful to know, from the outset, that a large portion of the complexity theory in *Theory of Computation* is disjoint from the material in other, related texts like Arora and Barak's *Computational Complexity: A Modern Approach*. The choice of which material to include is more complicated to judge without discussing the details, so — read on!

2 Summary

The material is based on Kozen's course, CS682: Theory of Computation, a semester course for first-year graduate students at Cornell University. In the first 270 pages, there are 41 primary lectures and another 10 supplementary or optional lectures interspersed throughout the text. In the sequel, there are 100 pages of homework exercises and (instructors be aware!) detailed solutions for *all* of the exercises.

I will begin by briefly highlighting the unique contributions of *Theory of Computation* with respect to the universe of complexity textbook material. Following that, there will be a general survey of the text, covering the essence of its total content.

2.1 What's unique in this book?

Let's begin with a few lecture titles: *Complexity of Decidable Theories. Applications of the Recursion Theorem. Complete Problems in the Arithmetic Hierarchy. The Friedberg-Muchnik Theorem. The Analytic Hierarchy. Fair Termination and Harel's Theorem.*

[6]

In many ways, the content of the book seems to be a reflection of a broader trend in complexity theory during the years of its creation: a general transition from logic-based complexity theory to more combinatorics-based complexity theory. As a result, both "worlds" are introduced in this book. In my opinion, it does an excellent job of discussing both – but more on that later.

2.2 Survey of Lectures

Note that the following grouping of lectures into sections is my own impression of the structure of the book based on what appear to be natural transitions in its focus or direction. The text itself is in fact 51 lectures in sequence, counting supplementary lectures.

2.2.1 Lectures 1-6: An Introduction

The 1st lecture by introducing the Turing Machine model. A proof of $\Omega(n^2)$ time for palindrome recognition on a one-tape TM follows. The 2nd lecture introduces all of the basic deterministic and nondeterministic space and time classes and their simple inclusions, as well as Savitch's theorem ($\mathsf{PSPACE} = \mathsf{NPSPACE}$). The 3rd lecture demonstrates the essence of known space hierarchies via padding techniques. The 4th lecture covers Immerman-Szelepcsényi's theorem (NSPACE is closed under complement for space $\geq \log n$). The 5th lecture introduces logspace computation and reducibility, and shows directed graph reachability complete for $\mathsf{NLOGSPACE}$. The 6th lecture proves the Cook-Levin theorem using *logspace* reductions (an interesting approach!).

2.2.2 Lectures 7-10: Alternation, PSPACE, and PH

Lecture 7 begins with a definition of an alternating Turing Machine and proves relationships between alternating and deterministic complexity classes $\big($e.g. $\mathsf{ASPACE}(S(n)) \subseteq \mathsf{DTIME}(2^{O(S(n))})\big)$. In the 8th lecture, complete problems for PSPACE are introduced, including satisfying quantified Boolean formulae and finding a forced win in chess. Following this, the 9th and 10th lectures are on the polynomial hierarchy, introducing oracles, building PH, and then relating levels of PH to oracle-based classes, e.g. $\mathsf{NP}^{\mathsf{NP}}$.

2.2.3 Lectures 11-12: Parallel Complexity and NC

Lectures 11 and 12 form a brief introduction to NC. Uniform families of circuits are defined, a family of logspace-uniform NC circuits to compute Boolean $n \times n$ matrix multiplication is given, and $\mathsf{NLOGSPACE} \subseteq \mathsf{Uniform\text{-}NC} \subseteq \mathsf{P}$ is proven.

2.2.4 Lectures 13-14: Probabilistic Complexity and BPP

The 13th lecture begins by briefly reviewing notions of probabilities of events, expectation, conditional probability, pairwise independence, and so on. Then, probabilistic Turing Machines are defined and in turn used to define the classes RP and BPP. An example of a useful, efficient probabilistic algorithm is given: namely, testing whether a low-degree multivariate polynomial with integer coefficients is identically 0 (where the straightforward deterministic algorithm requires exponential time). Then, in the 14th lecture, amplification is introduced (in the context of reducing probability of error exponentially with repetition) and used for a proof of $\mathsf{BPP} \subseteq \Sigma_2^{\mathsf{p}} \cap \Pi_2^{\mathsf{p}}$.

2.2.5 Lectures 15-20: IP and PCP

In lecture 15, IP is defined. The 16th and 17th lectures cover the proof of IP = PSPACE. In lecture 18, $\mathsf{PCP}(r(n), q(n))$ is defined. Then lectures 19 and 20 discuss a proof of $\mathsf{NP} \subseteq \mathsf{PCP}(n^3, 1)$. Two comments: (i) $\mathsf{NP} \subseteq \mathsf{PCP}(\log n, 1)$ is known; the book gives an intentionally simplified exposition – this is good – and (ii) the PCP proof in the text uses the original argument involving arithmetization, linearity testing, and so on, as opposed to Dinur's later combinatorial proof with expander graphs – this is unfortunate. (I feel that Dinur's proof is, if nothing else, easier to follow. However, it was published the same year that *Theory of Computation* was published.)

2.2.6 Lectures 21-27: Complexity of Decidable Theories and ω-Automata

The next series of lectures give a treatment of the complexity of first-order and second-order theories. The 21st lecture gives a proof showing that the first-order theory of dense linear order without endpoints is PSPACE-complete. The first-order theory of reals with addition and multiplication is shown to be decidable in lecture 22, and is shown to be NEXPTIME-hard in lecture 23. The 24th lecture shows the theory of integer addition to be complete for the class accepted by alternating Turing Machines with at most n alternations running in time $O\left(2^{2^{n^{O(1)}}}\right)$. The monadic second-order theory of successor is shown to be decidable, but not in elementary time (i.e. time bounded by a stack of exponentials), using Büchi, Rabin, and Muller automata over lectures 25-27.

2.2.7 Lectures 28-29: Relativization and Sparse Sets

The next lectures cover a couple of theorems directly relevant to the $\mathsf{P} \stackrel{?}{=} \mathsf{NP}$ question. Lecture 28 introduces Baker, Gill, and Solovay's proof that there are oracles A and B such that $\mathsf{P}^A = \mathsf{NP}^A$ and $\mathsf{P}^B \neq \mathsf{NP}^B$ as well as discusses the rise and fall of the Random Oracle Hypothesis – that containments or separations that hold with probability 1 with respect to a random oracle hold in the unrelativized case (which is particularly apt in light of the proof of IP = PSPACE earlier!). Lecture 29 introduces Fortune's and Mahaney's theorems – that sparse languages cannot be coNP-complete or NP-complete, respectively, unless P = NP.

2.2.8 Lectures 30-31: Circuit Lower Bounds

The aim of these next lectures, combined with a subsequent supplementary lecture, is to prove two complementary results. Lecture 30 sets up the beginning of a proof that that there exists an oracle A such that $\mathsf{PH}^A \neq \mathsf{PSPACE}^A$. Lecture 31 proves $\mathsf{PARITY} \notin \mathsf{AC}^0$. Supplementary Lecture H combines this result with Håstad's Switching Lemma to complete the proof from Lecture 30.

2.2.9 Lectures 32-34: The Gap Theorem and the Recursion Theorem

In the 32nd lecture, the Gap Theorem is stated (that there exist arbitrarily large recursive gaps in the complexity hierarchy). This leads into Blum's Speedup theorem – that there exist pathological, computable functions with no asymptotically optimal algorithm. In lecture 33, Gödel numbering is defined in order to prove the Recursion Theorem, which is in turn directly used to prove Rice's theorem in lecture 34 – that every nontrivial property of recursively enumerable languages is undecidable.

2.2.10 Lectures 35-41: The Arithmetic and Analytic Hierarchies

The 35th lecture introduces Turing reducibility in order to define the arithmetic hierarchy, and in lecture 36, various complete problems are demonstrated, e.g. $\{M : L(M) \text{ is recursive}\}$ is shown to be complete for Σ_3^0. Lecture 37 introduces Post's problem ("show that there are more than two Turing degrees"), and the 38th lecture resolves the problem using a finite injury priority argument. This leads toward a definition of the analytical hierarchy in lecture 39, and a proof of Kleene's theorem in lecture 40. Finally, lecture 41 gives an real-world application for the analytical hierarchy in fair termination of concurrent programs, showing that the decision problem is Π_1^1-complete.

3 Opinion

I really enjoyed reading *Theory of Computation*, and I think that had a lot to do with its structure. A huge advantage of having the material divided up into lectures instead of chapters is that you end up with bite-sized morsels of reading. It takes a nontrivial commitment of your time to fully read through a 20-25 page chapter of a textbook. On the other hand, you can go right through one of Kozen's 4-7 pages lectures in just 15-20 minutes, then put the book down and feel as if you've accomplished learning something. (It's hard to overstate how *awesome* that feels.)

As a result, I think the book definitely lends itself quite easily toward use in the classroom and even for self-learning. I imagine one could easily structure as a course around reading a "lecture" in the textbook prior to each in-class lecture over the same material. Since each chunk of reading is so short, you won't burn out your students with the constant reading.

One word of caution for prospective instructors: Since the answers for every exercise are in the text proper, you won't be able to give graded assignments directly out of the book. But is that necessarily a terrible situation? On the contrary, get creative!

On the topic of which material the book covers, I feel that its approach has some unique strengths and weaknesses. You can ask a question like, *should* a standard, graduate-level complexity course spend time covering, say, complete problems in the arithmetic hierarchy? There are always tradeoffs involved. Spend time on the arithmetic hierarchy, and you sacrifice time that could have been spent on derandomization, inapproximability, communication complexity, Barrington's theorem, or whatever else your favorite, hot topic might be.

On the other hand, Kozen's book does an excellent job of discussing many topics that are *not* in other texts. This material can be used to supplement any existing course in complexity theory at both the undergraduate and graduate levels.

Final verdict? Awesome lecture-style format. Very well-written explanations throughout the book. Definitely a useful, fun book.

Review of[7] of

Codes: an introduction to information communication and cryptography

by Norman L Biggs

Springer, 273 pages, paperback, 2008

Review by Robert J Low mtx014@coventry.ac.uk

1 Introduction

Coding is the representation of information originally expressed as a sequence of symbols by means of another sequence of symbols. There are three principal reasons for doing this:

1. Efficiency. We want to represent the information by as short a string as possible.
2. Robustness. We want to represent the information by a string in such a way that if the message is damaged, the corrupted message enables the reconstruction of the original message.
3. Security. We want to represent the information in such a way that unauthorized persons cannot read it.

This textbook aim to provide an introduction to coding as understood above. The three aspects are dealt with in turn, and the fundamental ideas of each one are explained clearly and with illustrative examples. Each section concludes with a useful collection of exercises, and there are suggestions for further reading at the end of each chapter. To assist the reader (especially the reader using the text for independent study) the author has provided answers to the odd-numbered exercises, ranging from complete worked solutions to brief hints.

2 Summary

Chapter 1: Coding and its uses

We begin at the beginning, with a discussion and overview of what is meant by coding, and why. Some historical context is provided, and the basic mathematical definitions are given here.

Chapter 2: Prefix-free codes

Clearly, in order for a code to be useful, it must be uniquely decipherable. The Kraft-McMillan inequalities are established, with the consequence that for any uniquely decipherable code, there is a prefix-free code with the same word lengths, and so we can limit the discussion to that of prefix-free codes without any reduction in efficiency.

Chapter 3: Economical coding

The memoryless source, $\mathcal{S}$, is now defined, and the notion of its entropy, $H(\mathcal{S})$ is introduced. Huffman's procedure for producing a compact prefix-free code is described, and proven to produce an optimal code. It is also shown that the average length of this code lies between $H(\mathcal{S})$ and $H(\mathcal{S}) + 1$.

Chapter 4: Data compression

In this chapter approaches to data compression are considered, in the context of stationary sources. In particular, arithmetic coding and the LZW algorithm are described.

Chapter 5: Noisy channels

The situation is now made slightly more realistic by the inclusion of a channel in the model. Naturally, the binary symmetric channel is the principal case considered. Conditional entropy is introduced, and employed in the consideration of channel capacity.

Chapter 6: The problem of reliable communication

We are now faced with the problem that the symbol stream leaving a channel may not be completely determined by the one entering it, because of channel noise. The general ideas of error-correction codes, namely Hamming distance, the minimal distance of a code and the relationship between the minimum distance and the number of errors which can be corrected are described.

Chapter 7: The noisy coding theorem

In this chapter Shannon's noisy coding theorem is described and discussed, though a full proof of the theorem is not provided here.

Chapter 8: Linear codes

The basics of linear codes are now considered. The use of generator matrices, parity check matrices, and syndrome decoding are described.

Chapter 9: Algebraic coding theory

The problem of decoding is still difficult in general linear codes. This chapter, then, shows how additional algebraic structure can be used to provide more useful codes. Cyclic codes are considered, and the special case of BCH codes over $\mathbb{Z}_2$, principally by means of particular examples.

Chapter 10: Coding natural languages

Real sources are not, of course, memoryless. Here we see some consideration of the properties of natural languages, and a bridge is provided to the study of cryptography in the remainder of the book.

Chapter 11: The development of cryptography

Now that cryptography has been introduced, a small discussion of the development of cryptography is provided. A general framework for discussing cryptosystems is given, and some classical cryptosystems, including Playfair and Vigenère, are described and discussed.

Chapter 12: Cryptography in theory and practice

Here, Shannon's general approach to cryptography and the notion of perfect secrecy are described. The one-time pad is described, DES is outlined, and AES is briefly referred to (but not described). As all systems up until now have been symmetric, there is clearly a problem of key distribution, and this is here given an abstract discussion, without any particular solution being considered yet.

Chapter 13: The RSA cryptosystem

We now meet the idea of public key cryptography. The system is give a clear description, including practical details of how the computations are carried out, and with plenty illustrative examples. The proof of correctness provided in the text is not quite complete, and the final details are an exercise for the student (but as it is an odd-numbered exercise, the solution is provided at the end of the book).

Chapter 14: Cryptography and calculation

In this chapter, a selection of topics all based on the common notion of calculations which are easy to carry out, but hard to invert (without insider knowledge) are described. After an introduction of the discrete logarithm problem, the El-Gamal cryptosystem and Diffie-Hellman key distribution are described. We are also given treatments of El-Gamal and RSA signature schemes.

Chapter 15: Elliptic curve cryptography

In this final chapter, the discussion is broadened out considerably. The use of finite groups as alphabets is introduced, and the general El-Gamal system is described in the context of elliptic curves.

3 Opinion

Overall, I found this to be an excellent introductory text, at the level of an advanced undergraduate or graduate student of computer science. The topics are well-chosen, and the exposition is rigorous while remaining clear. In places the notation and terminology deviate slightly from standard practice, but always in a way which is sensible, and without (or so it seems to me) placing serious difficulty in the way of a student investigating the literature. A study of this book provides a good introduction to the areas it covers, and prepares the student well for further, more detailed, investigation of information theory, error correcting codes, or cryptography.

I thoroughly enjoyed reading the book, and have already begun recommending it to my students.

Review of[8] of

Finite Fields and Applications
by Gary L. Mullen and Carl Mummert
Student Mathematical Library, Vol. 41
American Mathematical Society, 2007

Review by Jeffrey Shallit[9]

Every mathematician is familiar with the finite field $\mathbb{F}_p$ of p elements, where p is a prime. Less familiar, perhaps are the finite fields $\mathbb{F}_{p^n}$ of p^n elements, where $n \geq 2$ — indeed, I once heard a very good mathematician assert (incorrectly) that these fields are given by the integers modulo p^n. This is a shame, since the finite fields are simultaneously very well-behaved and still mysterious.

This little book is an introduction to finite fields, together with three chapters giving applications to combinatorics, algebraic coding theory, and cryptography. It is based on lectures notes used for an advanced undergraduate topics course.

Chapter 1 begins with the construction of finite fields. Here $\mathbb{F}_{p^n}$ is given as the splitting field of the polynomial $X^{p^n} - X$ over $\mathbb{F}_p$. To find out what a splitting field is, the reader needs to go to Appendix A, where some basic results about algebra and number theory are given, some without proof. Some examples of finite fields are given (by explicitly giving the multiplication tables). Additional sections cover the trace and norm, normal bases, irreducible polynomials, and permutation polynomials.

The fun begins in Chapter 2. Here the authors show how to apply ideas of finite fields to various problems in combinatorics: construction of latin squares, affine and projective planes, block designs, and Hadamard matrices. A *latin square* is an $n \times n$ array with entries in $\{0, 1, \ldots, n-1\}$ such that each row and column contains each symbol exactly once. Counting the number of distinct latin squares is a challenging problem, and the answer is only known for $1 \leq n \leq 11$. The authors remark, "no formula for l_n has been found and it is possible none exists".

[This is an opportunity for a brief rant about this kind of claim, which is often made in mathematical texts. I would like to say the claim is simply false. However, the real problem is that the term "formula" is so vague that it is not really clear what the authors mean. Making the term "formula" precise is, indeed, one of the great accomplishments of the theory of computation, and it's a real shame more mathematicians don't seem to appreciate this. As Herb Wilf has argued [2], the correct definition of "formula for x" is just "Turing machine to compute x". And, if we adopt that definition, then of course there *is* a formula for l_n, and we are left with the much more interesting question, how *efficiently* can we compute l_n?

Along the same lines, I recently attended a talk where a mathematician talked about computing the n'th digit of π "without computing the previous digits". I pointed out — to no avail — that he hadn't formalized the vague notion of "computing x without computing y", and it would be much more sensible to talk about computing the n'th digit of π using $O(\log n)$ space instead. Again, one of the great achievements of the theory of computation is making such a notion precise, but many mathematicians don't seem to appreciate this. Rant over.]

The authors next turn to orthogonal latin squares; these are two latin squares of the same order such that superimposing one on the other gives all n^2 pairs exactly once. A set of latin squares is *mutually orthogonal* if every pair has this property. As the authors show, a simple construction using finite fields produces $q-1$ mutually orthogonal latin squares of order q, if q is a prime power.

[9] School of Computer Science, University of Waterloo, Waterloo, ON N2L 3G1, Canada.

They also show how orthogonal latin squares are linked with finite projective planes.

The theory of block designs has resulted in many papers (*Mathematical Reviews* lists over 3000 papers on the subject). Quoting the book under review, "a *tactical configuration* is a set of v symbols arranged in b sets, called blocks, such that each block is of size k and each symbol occurs in exactly r blocks". And "a (v, k, λ) *balanced incomplete block (BIB) design* is a tactical configuration with v symbols and blocks of size k such that each pair of distinct symbols occurs in exactly λ blocks". These designs have applications to statistics and experiment design, and much work is devoted to determining for which triples (v, k, λ) a BIB design exists. (The book states that the existence of $(22, 8, 4)$ designs is open, but in fact, this was resolved negatively in 2007, the year of the book's publication [1].) The authors show how finite fields can be used to construct many block designs.

In the last section of Chapter 2, the authors show how finite fields can be used to construct Hadamard matrices. These are square matrices H, of dimension n and with entries in $\{-1, +1\}$, such that $HH^T = nI$, where I is the identity. However, the constructions depend on two theorems from a book of Lidl and Niederreiter, which are stated without proof.

Chapter 3 is devoted to algebraic coding theory. This is also a very active area of research, with applications to (among other things) data compression. Here the main topics are BCH codes (with proof again deferred to Lidl and Niederreiter), Goppa codes, perfect codes, and the relationship between codes, designs, and Latin squares.

The last chapter deals with cryptography. Here the main topics covered include RSA, Diffie-Hellman, the double-round quadratic cipher of Koblitz, elliptic curve cryptography, Dickson cryptosystems, and threshold schemes.

The book is generally well written, and I only found one typographical error [p. 74, "Let H_n is a Hadamard matrix of order n."] It could be used as the basis for a 1-semester course for advanced undergraduates or graduate students. Because the book often omits details of proofs, anyone teaching from the book will need to supplement it with other references. However, anyone reading this book will certainly gain an appreciation for the beauties and mysteries of finite fields, and their wide applicability to other areas of mathematics.

References

[1] R. Bilous et al., There is no $(22, 8, 4)$ block design, *J. Combin. Des.* **15** (2007), 262–267.

[2] H. Wilf, What is an answer?, *Amer. Math. Monthly* **89** (1982), 289–292.

Review of[10] of

The Life and Times of the Central Limit Theorem

by William J. Adams

American Math Society, Second Edition

Review by Miklós Bóna

Intuitively speaking, the Central Limit Theorem describes the behavior of the sum $S_n = X_1 + X_2 + \cdots + X_n$ of n mutually independent random variables, each of which has only a slight effect on the sum. The result is that if n is large, then the distribution of S_n is approximately normal. The name "Central Limit Theorem" was coined in 1920 by George Pólya.

In the world of precise mathematical statements, there is a large collection of theorems that are called Central Limit Theorems; they describe the conditions under which the conclusion discussed in the previous paragraph is valid. In this book, the author discusses the historical process from the first appearances of the Central Limit Theorem in its imprecise form to modern survey papers.

The story starts with Jacob Bernoulli, who, at the end of the seventeenth century, in his book *Ars Conjectandi*, discussed questions like the following one. Let us assume that a box contains 5000 pebbles, 3000 of which are white and 2000 of which are black. If we remove 500 pebbles, how sure can we be that the number of white pebbles removed is close to 300? He then proved an upper bound for the probability of the event that the number of white pebbles will be outside the interval $[290, 310]$. What is important about Bernoulli's work is that he was not interested in the *a priori* probability of an event, but instead he was interested in the level of precision at which results of experiments will approach that a priori probability.

The next significant figure in the development of the field was Abraham de Moivre, who, like Bernoulli, was interested in error term estimates. This quest led him to look for approximations for $n!$ since factorials regularly appear in arguments using involving choices of subsets. He essentially rediscovered Stirling's formula, that is, the fact that $n! \sim n^n e^{-n}\sqrt{2\pi n}$. His most frequently used technique involved computations of $\int e^{-x^2}$, often with power series.

As a consequence of De Moivre's work, by the end of the eighteenth century, the use of integrals of e^{-x^2} was well established. Besides a tool in the estimation of errors in probabilistic calculations, it was also used in mathematical astronomy. In fact, in 1732, the Academy of Sciences in Paris solicited papers explaining why planetary orbits were not completely circular and why they were closer to elliptical curves. No prize was awarded. However, two years later, the prize was given to John Bernoulli and his son Daniel. These two authors, in different papers, argued that the small differences between the orbits and circular curves could not be attributed to chance. Their work was continued by Pierre-Simon Laplace, who extended his focus to *cometary* orbits. There were 97 known comets at that time, and Laplace assumed that the orbit of each of them is the result of a variable of the same distribution. In modern form, we would say that he used a Central Limit Theorem for *identically distributed* mutually independent random variables.

Then comes a short chapter on the history of the concept that an error of observation is composed of elementary errors, and the effect of these elementary errors is diminished by their independence and multitude. This gets us to the *abstract* theory of the central limit theorem, and the three Russian mathematicians who brought that about, namely Pafnuty Lvovics Chebyshev, and his students Michail Vasilevich Lyapunov and Andrei Andreevich Markov. Their papers read reasonably close to contemporary research articles, as is illustrated by four papers of Lyapunov

[10]

that can be found in the Appendix. The inclusion of two of them is new to the second edition. Chebyshev started his career in the field by noticing that earlier work by Poisson lacked in rigor in its error estimates. Later Markov criticized one of Chebyshev's papers for the same reason. When Chebyshev could not provide a rigorous argument with the methods of classical analysis, he created the ingenious method of moments. This is why "Chebyshev's theorem" (the fact that it is very unlikely that a variable differs from its mean by a multiple of its standard deviation) was proved by him. It is worth pointing out that the other well-known result of the same kind, "Markov's theorem" (the fact that it is very unlikely that a random variable differs from its mean by a multiple of the mean" is also implicit in Chebyshev's work before its publication by Markov. Lyapunov turned to probability theory relatively late in his career, and worked exclusively with *discrete* random variables. We get an interesting peak into the life of pre-communism Russia through the lives of these three mathematician as public figures.

This ends the part of the book that was written by the author. Now comes the part devoted to the modern era. It starts with a 1945 survey paper of W. Feller on Fundamental Limit Theorems in Probability, and continues with a 1986 paper of L. LeCam that surveys the influential works about Central Limit Theorem that were written in the 1930s. It is remarkable how much more reader-friendly these papers are than their predecessors. The part is concluded by an entertaining comments-and-rejoinder section. The comments are from H. F. Trotter, J. L. Doob, and David Pollard, and the rejoinder is written by L. LeCam.

The book ends by the aforementioned inclusion of four papers of Lyapunov in the Appendix.

Historians of mathematics, some historians of Russia, and perhaps a few probabilists will find the book exciting. The average mathematician will probably read a few parts of it, but will find the countless versions of the Central Limit Theorem a little bit too many.

Miklós Bóna

Review of[11] of
Pearls of Discrete Mathematics
by Martin Erickson
CRC Press, 2010
280 pages, Softcover

Review by
Robert Szarka robert.szarka@uconn.edu
Department of Economics, University of Connecticut

1 Introduction

Given that a course in discrete mathematics often serves as a gateway to more advanced study, the subject will already be familiar to nearly all mathematicians and computer scientists, as well as most who have ventured beyond introductory calculus during their college careers. We might say simply that the field is about things that can be counted, but such a definition belies the many and varied objects of study it encompasses: logic, set theory, combinatorics, probability, number theory, algebra, graph theory, information theory, and more.

This motley mix of topics results in some thick introductory texts, but *Pearls of Discrete Mathematics* is not one of those. It could conceivably serve as an eclectic introduction to the field for a student already comfortable with reading and writing proofs, but it's probably more appropriate as the sequel to the typical introductory course in discrete math.

2 Summary

Aside from a familiarity with proofs and the basics of set theory, little exposure to discrete math is formally assumed in this book. The first two chapters even summarize the necessary material on permutations, combinations, and Pascal's triangle, though this brief presentation is unlikely to be enough for a reader with no prior exposure to the subject. Likewise, Chapter 10 presents key results in probability such as Bonferroni's, Markov's, and Chebyshev's inequalities, but in too cursory a manner for a first exposure to the topic.

The book's 24 chapters are divided into eight sections of three chapters each: "Counting: Basic"; "Counting: Intermediate"; "Counting: Advanced"; "Discrete Probability"; "Number Theory"; "Information Theory"; "Games"; and "Algorithms". Aside from the chapters mentioned above, whose results are used in subsequent chapters, each section is essentially independent. Instructors in search of a text for a topics course, or an independent reader with an interest in particular topics, will likely encounter few problems in skipping over the later chapters of a particular section or even entire sections.

The coverage of topics seems to be driven by the vagaries of the author's interests rather than by suitability for any particular area of application. Indeed, though much of the material—e.g. Markov chains, partitions of an integer, Shannon's theorems, or graph theory—would be of use in

diverse applications, Erickson makes no mention of them. Instead, chapters are devoted to examples such as rook and queen paths or solving Sudoku puzzles.

A substantial portion of the book is devoted to posing more than 300 problems from a wide range of difficulty. Solutions are given for all, with Mathematica code for solutions that require recourse to a calculator or computer.

3 Opinion

Pearls of Discrete Mathematics seems best suited for use in a topics course, where its modularity and wealth of solved problems will make it easy to integrate into a variety of curricula. I am less sanguine about its use for self-study, even for those with exposure to discrete math at the level of the usual introductory course. In some places, the presentation is quite thorough, to the point of offering multiple proofs for a given theorem; more than once, though, I found myself resorting to other books to flesh out the details. Despite this shortcoming, the abundance of solved problems may make it worthwhile even to those who find the level of detail insufficient to use it as their sole text.

Whether this book is enjoyable will probably depend on whether the reader is interested in discrete math for its own sake. As an economist with an admittedly utilitarian interest in the subject, I found the selection of topics frustrating. (Alas, the section labeled "Games" is not about game theory in the tradition of von Neumann and Morgenstern.) Even so, there was something to like in each section. Since the book's organization rewards selective reading, even those with only a casual interest may want to give it a look.

Review of[12]
Design Theory
by C. C. Lindner and C. A. Rodger
CRC Press, 2009
264 pages, HARDCOVER

Review by
Dimitris Papamichail, dimitris@cs.miami.edu
Dept. of Computer Science, University of Miami, USA

1 Introduction

Design theory is the part of combinatorics that deals with the existence and construction of block designs. Designs are collections of sets (or blocks) of objects (or symbols, points), with specified properties or features. In their general form, they can be defined by five parameters, the number of points, v, the number of blocks of points, b, the number of blocks containing a given point, r, the number of points in a block, k, and the number of blocks containing 2 (or more) points, λ. As such, block designs are also referred to as (v, b, r, k, λ)-designs, or (v, k, λ)-designs, since the parameters are not independent (and three can define the rest). Quite often additional constraints are placed to the designs, according to the defining problem.

As an example, we can consider the "Kirkman schoolgirl problem" which is mentioned in the book, where the following question if asked: Is it possible for a schoolmistress to take 15 schoolgirls on a walk each day of the 7 days of the week, walking with 5 rows of 3 girls each, in such a way that each pair of girls walks together in the same row on exactly one day? This is equivalent to asking whether a (15, 45, 7, 3, 1)-design exists, with the additional property of being able to partition the blocks into 7 *parallel* classes, each class being a partition of the set of points.

Design theory has been greatly influenced and directed by the design of experiments in a number of different areas, such as medicine, agriculture and manufacturing.

2 Summary

This book focuses mainly on construction techniques for basic block designs, such as Steiner triple systems, latin squares and finite projective and affine planes. It presents a number of important results and theorems with accompanied proofs.

The following designs are referenced extensively in the review and are summarized here:

- **STS – Steiner Triple System:** A (v, 3, 1)-design, where blocks are triplets and each pair of points appears in exactly one triplet.
- **PBD: Pairwise balanced design:** A (v, k, 1)-design, blocks can be of arbitrary size, but each pair of points appears in exactly one block.
- **KTS: Kirkman triple system:** As explained in the introductory example, an STS with parallel classes.

- **MOLs: Mutually orthogonal latin squares:** Latin squares that are pairwise orthogonal. Orthogonal latin squares have the property that when superimposed, the resulting ordered pairs are distinct.

The contents of each chapter are outlined below.

Chapter 1 This chapter examines the conditions for existence and several construction methods of Steiner triple systems (STS) of order v, sets of triplets drawn from a set of v symbols such that every pair of symbols appears in exactly one triplet. After a short proof that such designs exist only when $v \equiv 1$ *or* $3 \pmod 6$ and an introduction to quasigroups, the authors present the Bose construction for STS of order $v \equiv 3 \pmod 6$ using idempotent commutative quasigroups, and the Skolem construction for systems of order $v \equiv 1 \pmod 6$ using semi-idempotent commutative quasigroups. Steiner triplet systems are generalized to pairwise balanced designs (PBD), where blocks do not need to have size 3, in order to introduce a construction method for systems of order $v \equiv 5 \pmod 6$. The introduction of quasigroups with holes leads to another construction method for STSs (with holes), followed by the Wilson construction. The chapter continues with the definition and properties of cyclic STSs and the $2n + 1$ and $2n + 7$ recursive construction methods for STSs.

Chapter 2 Following the introduction of λ-fold triple systems, a generalization of the STSs where pairs of distinct elements belong now to exactly λ triplets, the authors introduce transversals and prove the existence of idempotent quasigroups of order n for all $n \neq 2$. They then present construction methods for 2-fold triple systems of order $3n$ and $3n + 1$ and Mendelsohn triple systems, in which triples can be ordered. Some comments on general λ-fold triple systems conclude the chapter.

Chapter 3 Quasigroups of order n are shown to be equivalent to *orthogonal* $n^2 \times 3$ arrays with certain properties and the notions of symmetric and semisymmetric quasigroups are introduced. Mendelsohn triple systems are shown equivalent to idempotent semisymmetric quasigroups and Steiner triple systems to totally symmetric quasigroups, results which lead to two new construction methods for the aforementioned systems.

Chapter 4 Since STSs of order n do not exist for certain values of n, maximum packings and minimum coverings can be used to construct block designs for these values, the former when we want to minimize the number of pairs of elements that are excluded from the triples (the *leave*) and the latter when we minimize the number of pairs that appear in more than one triplet (the *padding*). Based on the value of $n \bmod 6$, which defines the size of the leave and padding, three different construction methods for each of the maximum packing and minimum coverings are presented.

Chapter 5 Kirkman triple systems (KTS) are introduced as resolvable STSs that can be partitioned into *parallel* classes, and PBDs of order $3n + 1$ are used to construct KTSs of order $6n + 3$. Several techniques to build PBDs with limited number of block sizes using mutually orthogonal latin squares and group divisible designs are discussed.

Chapter 6 In this chapter the authors formally define mutually orthogonal latin squares (MOLs) and conditions for their existence. They present a method to construct them using finite

fields and as direct products of MOLs. They continue to disprove the MacNeish and Euler conjectures about MOL existence using counter examples and show a general construction method for MOLs of order n, for all $n \neq 2$ *or* 6.

Chapter 7 The seventh chapter introduces affine and projective planes as PDBs where blocks are represented by lines and have certain properties. These planes are shown to be related and equivalent to complete sets of MOLs.

Chapter 8 Two Steiner triple systems are disjoint when they do not have any triples in common and isomorphic when the set of points of one is a permutation of the other's. To prove that there always exists an isomorphic disjoint mate for every STS, Teirlinck devised a recursive algorithm using a series of transpositions. The authors also present the more general case when pairs of STSs share x triples and determine the values of x when this is possible, demonstrating another construction method.

Chapter 9 Incomplete latin rectangles have the property that every symbol occurs once on each column and row. The authors in this chapter examine the necessary and sufficient conditions for such rectangles to be embedded in latin squares and provide a proof using a reduction to edge colored bipartite graphs. Then they examine the conditions for embedding incomplete idempotent latin squares in complete ones and move on to partial Steiner triple systems embeddings.

Chapter 10 The last chapter deals with Steiner quadruple systems. As with the first chapter, a variety of methods to construct these systems are presented, together with proofs of the necessary and sufficient conditions for such systems to exist.

3 Opinion

Design Theory by Lindner and Rodger is exploring an area in combinatorics that concerns the conditions for existence of several important block designs and methods to construct them. It explains these techniques in good detail and accompanies the theory with helpful illustrations that serve their purpose admirably. The material of this book has been carefully ordered and results, theorems and constructions build successively on previously presented definitions and explanations. Limited background is required for comprehending the theory, which, as a result, is accessible to a wide mathematically oriented audience, both undergraduate and postgraduate. A few non obvious prerequisites are mentioned explicitly in the text when required, although concepts such as *finite fields* could actually be devoted a few pages and explained in detail, instead of being just mentioned as prerequisites.

Although the material is interesting and presented effectively, the book could greatly benefit from examples and applications, especially since design theory has been inspired and directed by the design of experiments in applied fields. Practitioners may find the use of this book somewhat cumbersome, if the goal is to determine which block design and construction method may be appropriate for a specific experimental design. Currently the whole book has a total of three examples of real problems that have influenced some of these techniques to be developed. In conjunction with a somewhat messy structure and the unclear separation of examples, comments, exercises and theorems, the use of this book as a reference is severely hindered. Having said that,

it is clear that the authors did not intend this text to be used as a cookbook or reference, but that is a shame, since only a few improvements, such as a more carefully structured and overviewed content, a greater number of applications/examples and a slightly better modularity, would go a long way to serve both theoretically and practically inclined audiences. Lastly, an introduction, overview of the field and summary of the results would also add to the value and readability of the book.

Overall, this text is a helpful addition to any combinatorist's library and provides a nice introduction to block design construction techniques, with a collection of carefully selected and effectively presented topics.

Review of[13] of
An Introduction to the History of Algebra
Solving Equations from Mesopotamian Times to the Renaissance
by Jacques Sesiano
Published by the AMS, 2009
174 pages, Softcover, $57.00 on amazon, $28.00 for AMS members at AMS

Review by
William Gasarch gasarch@cs.umd.edu

1 Introduction

When reading a history of math book I usually have two contradictory thoughts:

- These people were really sharp to be able to solve that problem given the notation and knowledge they had to work with.
- Why are they solving this problem in such a clunky way? Can't they do better?

Reading this book I had the same thoughts. I'll add a third one: *That is an odd problem to work on!*

The book is not a complete history of Algebra in the time given in the title. This is good– that would be too long and tedious. This is a book that can be read by a high school student. She may well be, as I was, both impressed and depressed by the mathematics she sees.

One very nice feature of the book is that there are many worked out examples of the methods they use. This is informative and also makes us thank our lucky stars that we have different methods.

2 Summary of Contents

The first chapter is on *Algebra in Mesopotamia.* The problems are all rather practical having to do with areas of fields of grain. Many of them become what we would call two linear equations in two variables. For these a common method was to let $z = (x + y)/2$ and $w = (x - y)/2$ and solve the problem with z, w rather than x, y. This is cumbersome but impressive for its day.

They also solved more complicated equations. The author states that its hard on the basis of just one old text to tell if there is a method or if it is trial-and-error. In most cases there does seem to be a method.

Sample Problem: I added the area of my two squares; (it gives) 1300. The side of one exceeds the side of the other by 10. (implicit: Find the size of both fields).

The knew the quadratic formula but since they didn't have the minus sign they have several cases: (1) $ax^2 + bx + c = 0$, (2) $ax^2 + bx = c$, (3) $ax^2 = bx + c$, (4) $ax^2 + c = bx$.

The second chapter is on *Algebra in Ancient Greece.* There are many problems about right triangles. So many that it seems that these problems were of interest independent of applications.

[13] ©2012, William Gasarch

Sample Problem: For a right-angled Triangle in which the measure of feet of the hypotenuse is 25 feet (and) the area 150 (square) feet, tell the height and the base separately.

The book also talks about Diophantus who was a rather isolated figure (mathematically). Nobody else worked on the kinds of problems he worked on in his day (seeking fraction solutions). The book *Arithmetica* by Diophantus is actually a set of problems and worked out solutions from which we can derive his methods.
Sample Problem: To find two square numbers having a given difference. Note that this is a rather abstract problem. Diophantus wants a general method.

The third chapter is on *Algebra in the Islamic World*. They worked on linear and quadratic problems (and sometimes higher degree) using geometric methods. They would interpret the problem geometrically in terms of areas of squares and parts of the square. Very clever! They also introduced a new type of problem where they would have two conditions:

- $x + y = c$ where c is some constant.
- $f(x, y) = d$ where d is some constant. Note that f could be rather general

Sample Problem:

$$\begin{aligned} x + y &= 10 \\ \frac{x}{y} + \frac{y}{x} &= 2 + \frac{1}{6} \end{aligned}$$

The remaining chapters on *Algebra in Medieval Europe* and *Algebra in the Renaissance* take the story up to a form we can recognize it. The last chapter has the cubic and quartic equations and imaginary numbers.

3 Opinion

The book is very good at what it set out to do: Present the history of algebra in a way that is accurate but not tedious. Note that the book does not have much in the way of political history or biography. It really is a history of Algebra.

The math needed to read this book is fairly elementary. A high school student who has had elementary algebra and has some maturity could read it. A Math PhD could also learn from it since this material is not taught often. The audience is not determined by how much math the reader knows. It is determined by if you are interested in the subject and willing to work some examples. If so, this book will benefit you.