PEARSON SERIES IN COMPUTER NETWORKING AND DISTRIBUTED SYSTEMS

THIRD EDITION

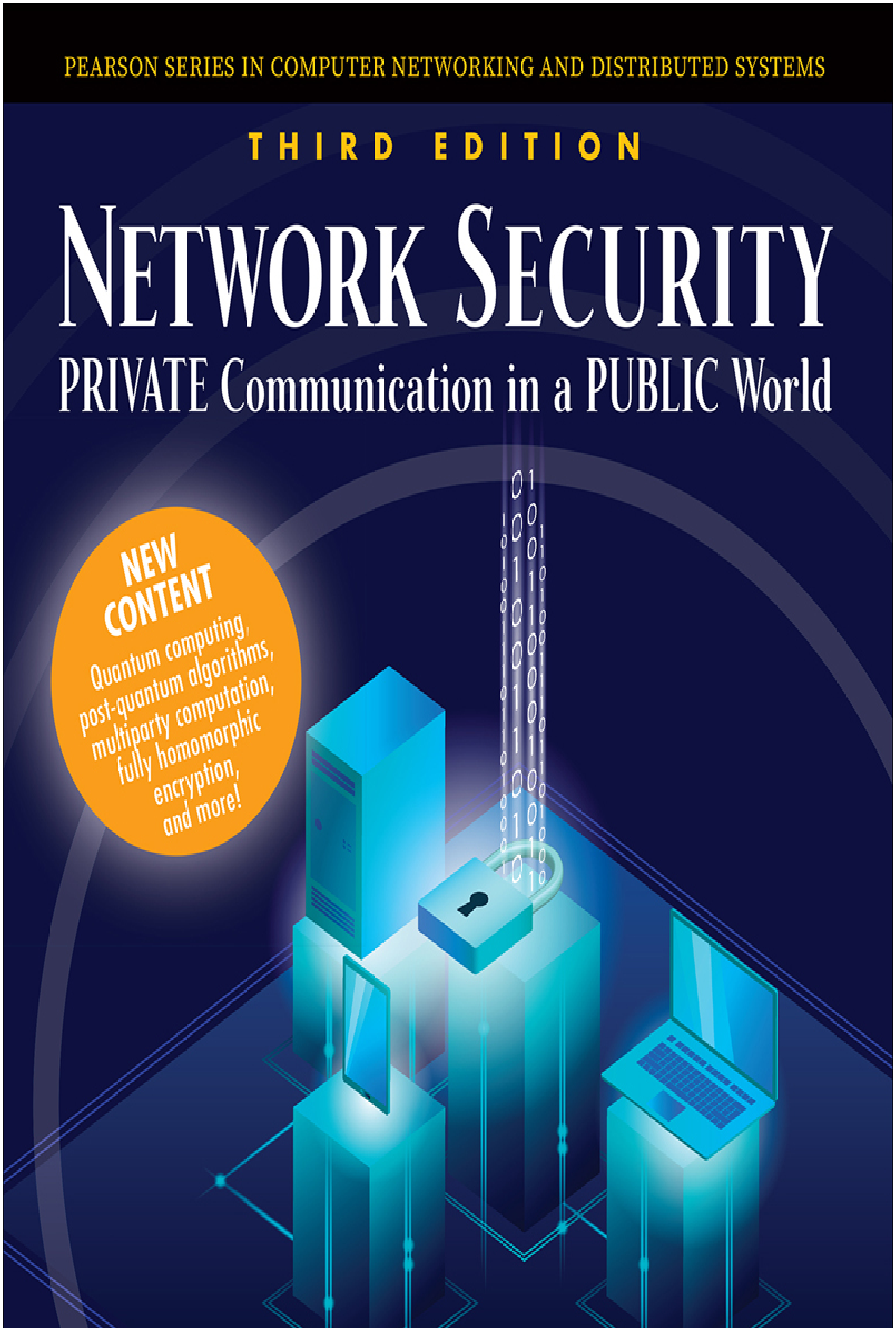
NETWORK SECURITY

PRIVATE Communication in a PUBLIC World

NEW CONTENT

Quantum computing, post-quantum algorithms, multiparty computation, fully homomorphic encryption, and more!

CHARLIE KAUFMAN • RADIA PERLMAN
MIKE SPECINER • RAY PERLNER



PEARSON SERIES IN COMPUTER NETWORKING AND DISTRIBUTED SYSTEMS

THIRD EDITION

NETWORK SECURITY

PRIVATE Communication in a PUBLIC World

NEW CONTENT

Quantum computing, post-quantum algorithms, multiparty computation, fully homomorphic encryption, and more!

Charlie Kaufman • Radia Perlman
Mike Speciner • Ray Perlner

About This eBook

ePUB is an open, industry-standard format for eBooks. However, support of ePUB and its many features varies across reading devices and applications. Use your device or app settings to customize the presentation to your liking. Settings that you can customize often include font, font size, single or double column, landscape or portrait mode, and figures that you can click or tap to enlarge. For additional information about the settings and features on your reading device or app, visit the device manufacturer's Web site.

Many titles include programming code or configuration examples. To optimize the presentation of these elements, view the eBook in single-column, landscape mode and adjust the font size to the smallest setting. ...

Network Security
PRIVATE Communication in a PUBLIC World

Third Edition

Charlie Kaufman • Radia Perlman
Mike Speciner • Ray Perlner

Addison-Wesley

Boston • Columbus • New York • San Francisco • Amsterdam • Cape TownDubai • London • Madrid • Milan • Munich • Paris • Montreal • Toronto • Delhi • Mexico CitySão Paulo • Sydney • Hong Kong • Seoul • Singapore • Taipei • Tokyo

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and the publisher was aware of a trademark claim, the designations have been printed with initial capital letters or in all capitals.

The authors and publisher have taken care in the preparation of this book, but make no expressed or implied warranty of any kind and assume no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information or programs contained herein.

For information about buying this title in bulk quantities, or for special sales opportunities (which may include ...

Pearson’s Commitment to Diversity, Equity, and Inclusion

Pearson is dedicated to creating bias-free content that reflects the diversity of all learners. We embrace the many dimensions of diversity, including but not limited to race, ethnicity, gender, socioeconomic status, ability, age, sexual orientation, and religious or political beliefs.

Education is a powerful force for equity and change in our world. It has the potential to deliver opportunities that improve lives and enable economic mobility. As we work with authors to create content for every product and service, we acknowledge our responsibility to demonstrate inclusivity and incorporate diverse scholarship so that everyone can achieve their potential through learning. As the world’s ...

Si spy net work, big fedjaw iog link kyxogy

Contents

Acknowledgments

Despite the controversies that crop up around security issues, it has been our experience that people in the security community are generally generous with their wisdom and time. It's always a little scary thanking specific people, for fear we'll leave someone out, but leaving everyone out seems wrong. It's not even the fair thing to do, since some people would be more egregiously wronged by being left out than others.

We'd like to thank the reviewers of various chapters of this book, which include Luís T. A. N. Brandão, Bill Burr, Lisa Carnahan, Lily Chen, David Cooper, Joan Daemen, Michael Davidson, Morrie Dworkin, Ned Freed, Simson Garfinkel, Jonathan Katz, John Kelsey, Yi-Kai Liu, Kerry McKay, Carl Miller, Dustin Moody, Nicky ...

About the Authors

Charlie Kaufman is currently Security Architect for Dell Storage Systems. Previously, he was the Security Architect for Microsoft Azure and before that for Lotus Notes. He has contributed to a number of IETF standards efforts including IPsec, S/MIME, and DNSSEC and served as a member of the Internet Architecture Board. He served on the National Academy of Sciences expert panel that wrote the book *Trust In Cyberspace*.

Radia Perlman is currently a Fellow at Dell Technologies. She is known for her contributions to bridging (spanning tree algorithm), routing (link state routing), and security (distributed systems robust despite malicious participants). She’s the author of *Interconnections: Bridges, Routers, Switches, and Internetworking ...*

1 Introduction

It was a dark and stormy night. Somewhere in the distance a dog howled. A shiny object caught Alice's eye. A diamond cufflink! Only one person in the household could afford diamond cufflinks! So it was the butler, after all! Alice had to warn Bob. But how could she get a message to him without alerting the butler? If she phoned Bob, the butler might listen on an extension. If she sent a carrier pigeon out the window with the message taped to its foot, how would Bob know it was Alice that was sending the message and not Trudy attempting to frame the butler because he spurned her advances?

That's what this book is about. Not much character development for Alice and Bob, we're afraid; nor do we really get to know the butler. But ...

2 Introduction to Cryptography

2.1 Introduction

The word *cryptography* comes from the Greek words κρυπτο (*hidden* or *secret*) and γραφη (*writing*). So, cryptography is the art of secret writing. More generally, people think of cryptography as the art of mangling information into apparent unintelligibility in a manner allowing a secret method of unmangling. The basic service provided by cryptography is the ability to send information between participants in a way that prevents others from reading it. In this book, we will concentrate on the kind of cryptography that is based on representing information as numbers and mathematically manipulating those numbers. This kind of cryptography can provide other services, such as

- integrity checking—reassuring ...

3 Secret Key Cryptography

3.1 Introduction

Secret key encryption schemes require that both the party that does the encryption and the party that does the decryption share a secret key. We will discuss two types of secret key encryption schemes:

- **block cipher**. This takes as input a secret key and a plaintext block of fixed size (older ciphers used 64-bit blocks, modern ciphers use 128-bit blocks). It produces a ciphertext block the same size as the plaintext block. To encrypt messages larger than the blocksize, the block cipher is used iteratively with algorithms called *modes of operation* that are the subject of the next chapter. A block cipher also has a decryption operation that does the reverse computation.
- **stream cipher**. This uses the key ...

4 Modes of Operation

4.1 Introduction

We’ve covered how to encrypt a 128-bit block with AES, or a 64-bit block with DES, and these primitives have the nice property that if an attacker changes any part of the ciphertext, the result of a decryption will be effectively a random number. Sadly, most useful messages are longer than 128 bits. **Modes of operation** are techniques for encrypting arbitrary-sized messages using the block encryption algorithms as primitives to be applied iteratively. There are additional desirable properties that such algorithms can have. If we send the same message multiple times, it would be desirable to have the encrypted message be different each time so that an eavesdropper can’t tell that we’re sending the same message ...

5 Cryptographic Hashes

5.1 Introduction

A hash function inputs an arbitrary-sized bitstring and outputs a fixed-size bitstring, ideally so that all output values are equally likely. **A cryptographic hash** (also known as a **message digest**) has some extra security properties:

- **preimage resistance**: It should be computationally infeasible to find a message that has a given pre-specified hash.
- **collision resistance**: It should be computationally infeasible to find two messages that have the same hash.
- **second preimage resistance**: It should be computationally infeasible to find a second message that has the same hash as a given message.

The term *message digest* was originally more popular, but *hash* is more commonly used today. As evidence that the world ...

6 First-Generation Public Key Algorithms

6.1 Introduction

This chapter describes the most common public key algorithms that have been deployed as of 2022 and that will probably remain the most common public key algorithms in use for several years. As we will show in Chapter 7 *Quantum Computing*, if someone could build a quantum computer of sufficient size, the algorithms in this chapter would no longer be secure. The world will soon be converting to different public key algorithms (see Chapter 8 *Post-Quantum Cryptography*). But the current algorithms, the focus of this chapter, are widely deployed and fascinating to understand.

Public key algorithms are a motley crew. All the hash algorithms do the same thing—they take a message and perform an ...

7 Quantum Computing

7.1 What Is a Quantum Computer?

Quantum mechanics predicts that it should be possible to build a computer that can do certain calculations much faster than would be possible on a conventional (classical) computer. Aspects of quantum mechanics may seem nonintuitive, but all evidence supports it. In this chapter, we describe how a quantum computer differs from a classical computer and give intuitive descriptions of the quantum algorithms most relevant to cryptography.

There are entire books about quantum mechanics. Our goal isn’t to pack years of physics and math into a few pages but to give some insight into the concepts, terminology, and notation, as well as the algorithms that run on quantum computers. And for readers who ...

8 Post-Quantum Cryptography

As we described in Chapter 7 *Quantum Computing*, a sufficiently large quantum computer implementing Shor’s algorithm would break our currently deployed public key algorithms. However, long before that can happen, the world will (hopefully) have converted to replacement algorithms. The replacement algorithms will be based on math problems that (hopefully) not even a combination of classical and quantum computers would be able to solve in a reasonable amount of time.

These new algorithms are known by several equivalent names: quantum-resistant, quantum-safe, or post-quantum cryptography (PQC). The world seems to have settled on the term *post-quantum*, so that is what we will use, even though we have noticed that the term ...

9 Authentication of People

> *Humans are incapable of storing high-quality cryptographic keys and they have unacceptable speed and accuracy when performing cryptographic operations. They are also large, expensive to maintain, difficult to manage, and they pollute the environment. It is astonishing that these devices continue to be manufactured and deployed, but they are sufficiently pervasive that we must design our systems around their limitations.*
>
> —Radia Perlman

Authentication is the process of reliably verifying the identity of someone (or something). There are lots of examples of authentication in human interaction. People who know you can recognize you based on your appearance or voice. A guard might authenticate you by comparing you with ...

10 Trusted Intermediaries

10.1 Introduction

If nodes Alice and Bob want to be able to communicate securely, they need to know keys for each other. Configuring each node with keys for every other node will not scale beyond a small number, so a **trusted third party** (someone that Alice and Bob trust) is used for introducing Alice and Bob to each other.

In this chapter we describe different types of systems based on trusted third parties. One is a system that uses only secret keys, in which case the trusted third party is usually known as a **KDC** (**Key Distribution Center**). In a public key system, the trusted third party is usually known as a **certification authority** (**CA**), and it signs **certificate**s, which assert things such as the mapping between the ...

11 Communication Session Establishment

Knock Knock!
Who’s there?
Alice.
Alice who?

…and you’ll have to read on to find secure ways of continuing…

This chapter analyzes various considerations when designing real-time communication handshakes. We start with very simple example handshakes that do authentication only (rather than also creating a session key and cryptographically protecting the data). These types of protocols are useful for simple scenarios, such as opening a door, and were common when people just wanted to replace sending a password in the clear with the least amount of effort. Even though most Internet communication today is done with TLS, it is still instructive to start with analysis of very simple handshakes. The second half ...

12 IPsec

As we said in Chapter 11 *Communication Session Establishment*, IPsec is a secure session protocol that runs on top of network layer 3 (see§11.7 *What Layer?*). The implication of running directly on layer 3 (*e.g.*, IP) is that each packet is independently cryptographically protected. IPsec does not guarantee that all packets will arrive or that those that do arrive will be delivered in the order they were sent. IPsec only guarantees that packets that do not meet the integrity check will be discarded, and packets that are duplicates will be discarded. This design makes it easy to implement in network adapters. IPsec does not need to buffer packets. IPsec can process and deliver packets independently, even if they arrive out of order. IPsec ...

13 SSL/TLS and SSH

The concepts in TLS (Transport Layer Security) have been covered in Chapter 11 *Communication Session Establishment*, and the concepts are similar to IPsec. Alice and Bob authenticate and establish cryptographic keys for the session.

TLS grew out of Netscape's SSL (Secure Sockets Layer) protocol. When the IETF took it over to improve and standardize it, they renamed it TLS (Transport Layer Security). Since being called TLS, it has gone through three revisions, so the latest version is 1.3 (RFC 8446). There is no real logic to why the TLS versions were named TLS 1.0, TLS 1.1, TLS 1.2, and TLS 1.3 rather than TLS version 1, TLS version 2, TLS version 3, and TLS version 4 or even why TLS 1.0 was not named SSL version 4. Most of ...

14 Electronic Mail Security

The first thing that springs to mind with the phrase *email security* is a message from Alice to Bob signed by Alice with her private key and encrypted with Bob’s public key. Lotus Notes had a proprietary, widely deployed implementation of encrypted email in 1989. PGP was created and released as open source by Phil Zimmermann in 1991. Various email standards for user-to-user signed and encrypted email were developed over twenty years ago, including

- PEM [RFC 1421—Feb 1993]
- PGP/GPG [RFC 2015—October 1996]
- S/MIME [RFC 2311—March 1998]

In the early 1990s, there were deployment barriers such as export controls and patents, but these have been resolved. So it is somewhat astonishing that none of these, or anything implementing ...

15 Electronic Money

Money has a long and storied history as a medium of exchange and a mechanism for measuring and storing wealth. Nations control the minting of coins, the printing of paper currency, and the quantity of money in circulation, trying hard through physical means to avoid counterfeit bills and coins. Nowadays, almost all but the smallest payments are made using credit cards, checks, and electronic funds transfers. Large cash transactions (suitcases full of money) are usually used only for illicit or illegal transactions.

Forms of electronic money have been common for many years. The Internet is used to manage bank accounts, make purchases with credit cards, do wire transfers, and pay bills.

New protocols for dealing with money, ...

16Cryptographic Tricks

We’ve covered the major cryptographic building blocks, such as secret key encryption and integrity checks, public key encryption and signatures, and hashes/message digests. In this chapter we’ll talk about some other functions. We won’t go into great detail about the math, or proofs, but we will at least demystify what these functions are attempting to do, give some intuition about how they accomplish it, and give examples of those that are used in actual real-world protocols.

16.1 Secret Sharing

Secret sharing is a way for someone, say, Alice, to store a piece of information, say, *S*, that must be kept secret from all except Alice but must also be retrievable by Alice. There is a tradeoff between robustly storing *S* so ...

17 Folklore

> *Whenever I made a roast, I always started off by cutting off the ends, just like I'd seen my grandmother do. Someone once asked me why I did it, and I realized I had no idea. It had never occurred to me to wonder. It was just the way it was done. Eventually I remembered to ask my grandmother. "Why do you always cut off the ends of a roast?" She answered "Because my pan is small, and otherwise the roasts would not fit."*
>
> —anonymous

Many things have become accepted security practice. Most of these are to avoid problems that could be avoided in other ways if you really knew what you were doing. It's fine to get in the habit of doing these things, but it would be nice to know at least *why* you're doing them. A lot of these issues have ...

Glossary

access control—a mechanism for limiting use of some resource to authorized users.

access control set—a synonym for access control list; some people make the distinction that the order of entries in an access control set cannot be significant, while the order of entries in an access control list might be.

ACL (access control list)—a data structure associated with a resource that specifies the authorized users.

active attack—one in which an attacker does something other than simply eavesdropping, for instance, transmits data, modifies data, or subverts the system so that it can impersonate an address.

ancilla—a qubit that is used by a quantum computation but which is initialized to a value independent of the computations's input.

ANSI— ...

Math

This supplement was written solely by me$_3$ (Mike Speciner) with the intention of providing material to expand and deepen your knowledge of the mathematics and algorithms behind many of the techniques explained in the book. You should be able to understand the rest of the book without this knowledge, but I$_3$ think it could still be helpful, or at least enlightening.

This supplement, including the homework problems: https://github.com/ms0/docs/blob/main/math.pdf.

M.1 Introduction

Computing technology has encouraged us to turn pretty much everything into bits. We use bits to represent numbers, text, images, audio, video, objects, scents, and even more abstract concepts like spacetime, quantum fields, and money. For a few of these, the representation ...

Bibliography

AGRA04 Agrawal, M., Kayal, N., Saxena, N., “PRIMES is in P”, *Annals of Mathematics*, Vol. 160 #2, September 2004, pp. 781–793.

ALAG93 Alagappan, K., *Telnet Authentication: SPX*, RFC 1412, January 1993.

BALE85 Balenson, D., “Automated Distribution of Cryptographic Keys Using the Financial Institution Key Management Standard”, *IEEE Communications*, Vol. 23 #9, September 1985, pp. 41–46.

BALE93 Balenson, D., *Privacy Enhancement for Internet Electronic Mail: Part III: Algorithms, Modes, and Identifiers*, RFC 1423, February 1993.

BELL74 Bell, D. E. and LaPadula, L. J., *Secure Computer Systems: Mathematical Foundations and Model*, M74-244, Mitre Corp., October 1974.

BELL90 Bellovin, S. M. and Merritt, M., “Limitations of the Kerberos Authentication ...

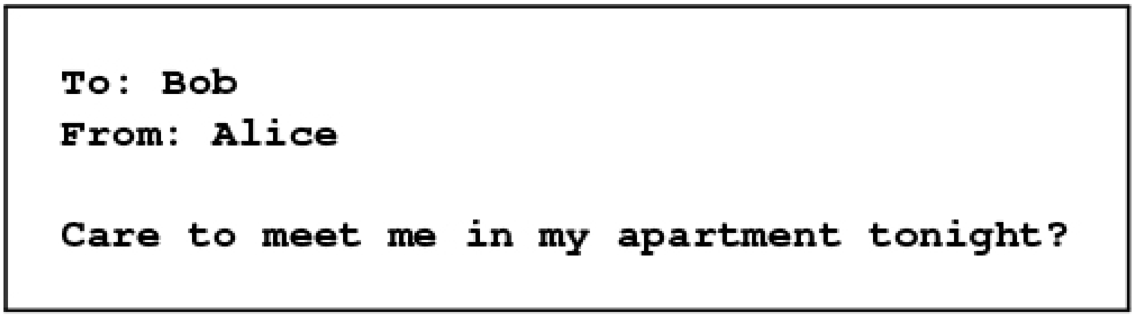
Index

Photo by marvent/Shutterstock

Code Snippets

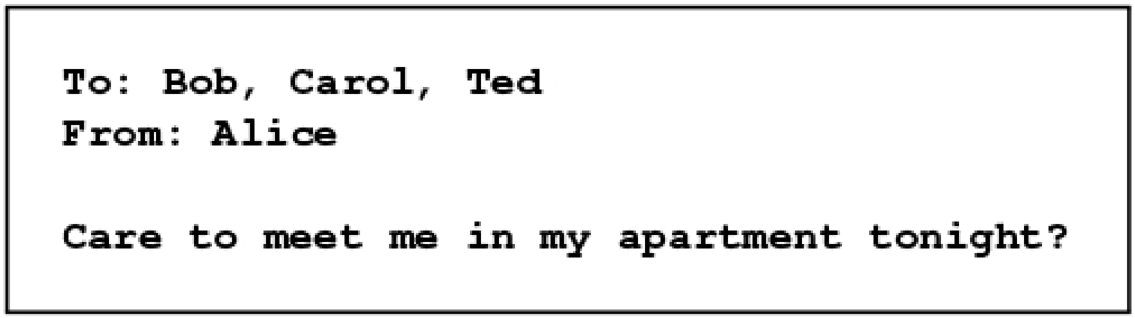
Many titles include programming code or configuration examples. To optimize the presentation of these elements, view the eBook in single-column, landscape mode and adjust the font size to the smallest setting. In addition to presenting code and configurations in the reflowable text format, we have included images of the code that mimic the presentation found in the print book; therefore, where the reflowable format may compromise the presentation of the code listing, you will see a “Click here to view code image” link. Click the link to view the print-fidelity code image. To return to the previous page viewed, click the Back button on your device or app.

```
Tacker, Ann A.      System Security Officer          44,122.10
|  m1   |  m2   |  m3   |  m4   |  m5   |  m6   |  m7   |  m8   |
Tacker, Ann A.      System Security Offi!z°Œ(%9™       54,122.10
|  m1   |  m2   |  m3   |  m4   |  m5   |  m6   |  m7   |  m8   |
```



To: Bob
From: Alice

Care to meet me in my apartment tonight?



To: Bob, Carol, Ted
From: Alice

Care to meet me in my apartment tonight?

```
def gcd(a,b) :
  while b != 0 :
    a,b = b,a%b    # replace <a,b> with a smaller pair with the same gcd
  return a
```

```
def xgcd(a,b) :       # Let A,B be the initial (input) values of a,b
  u,v,w,x = 1,0,0,1   # then a is u*A + v*B and b is w*A + x*B throughout
  while b != 0 :
    q,r = divmod(a,b) # quotient and remainder
    a,u,v,b,w,x = b,w,x,r,u-q*w,v-q*x
  return a,u,v
```

```
def pow(x,n) :
  if n == 0 : return 1
  z = 1 << n.bit_length() >> 2
  y = x             # initialize the result-so-far for the leftmost exponent bit
  while z :         # while there are still exponent bits to process
    y *= y          # square the result-so-far
    if n&z : y *= x # if the exponent bit is 1, multiply result-so-far by x
    z >>= 1         # move on to the next exponent bit
  return y
```

```
append an n by n identity matrix to the right of M, making M n by 2n
for c in Range(n):         # for each column c, 1 through n
  for r in Range(c,n):     #   look in row c and rows below
    if M[r,c]: break;      #     until find a nonzero element in column c
  else:                    #   no suitable row found
    raise ZeroDivisionError(‘not invertible’)
  apply S(r,c);            #   swap the suitable row with row c
  apply T(1/M[c,c],c);     #   make the diagonal element 1
   for r in Range(c+1,n) : #   for each row below row c
     apply A(-M[r,c],c,r); #     zero out column c
for c in Range(2,n):       #   for each column c, 2 through n
  for r in Range(c-1):     #     for each row above row c
    apply A(-M[r,c],c,r);  #       zero out column c
```