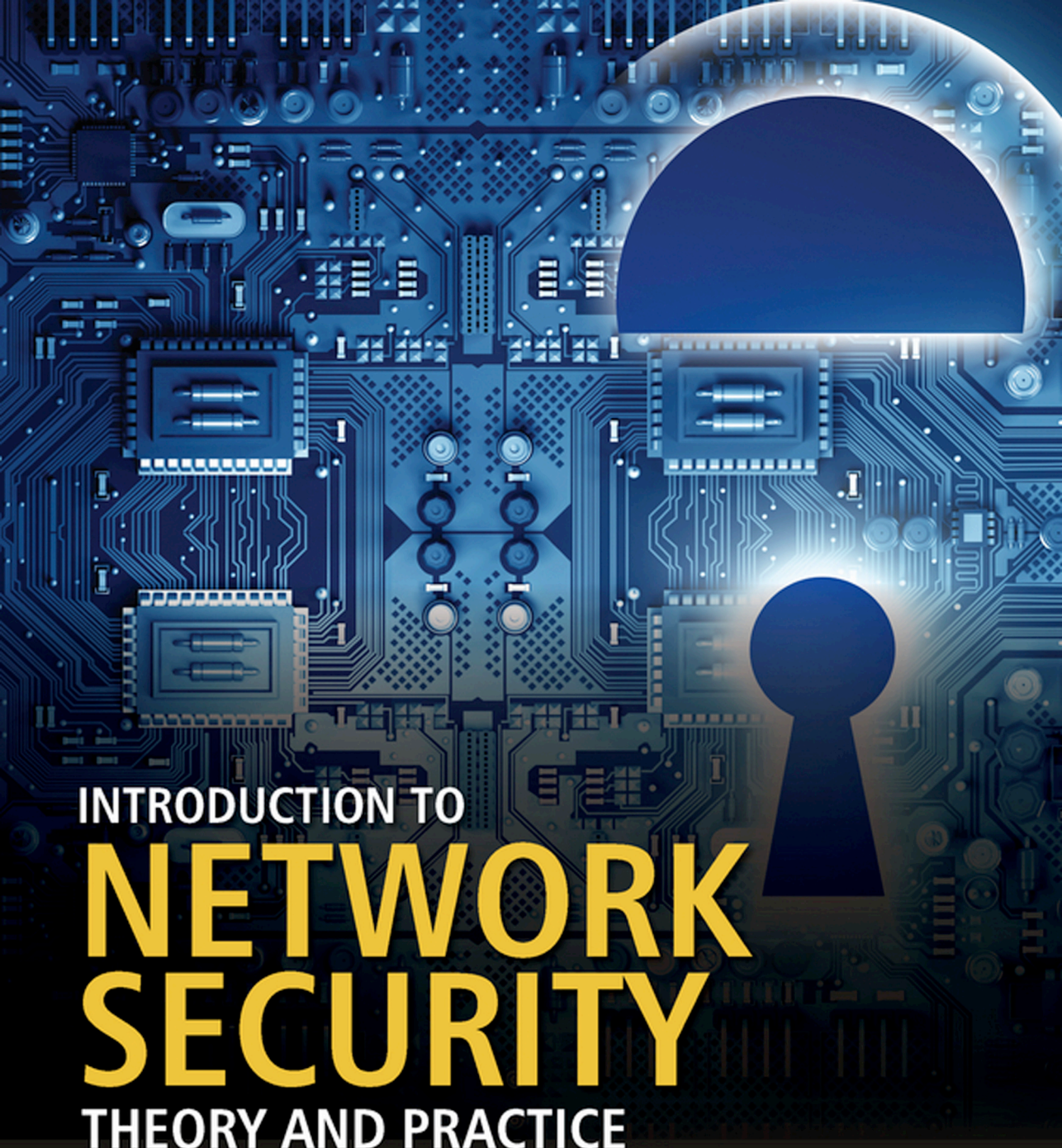

INTRODUCTION TO NETWORK SECURITY

THEORY AND PRACTICE

JIE WANG • ZACHARY KISSEL

Higher Education Press

WILEY

INTRODUCTION TO NETWORK SECURITY

INTRODUCTION TO NETWORK SECURITY

THEORY AND PRACTICE

Jie Wang

University of Massachusetts Lowell, US

Zachary A. Kissel

Merrimack College, US

WILEY

This edition first published 2015

Published by John Wiley & Sons Singapore Pte Ltd, 1 Fusionopolis Walk, #07-01 Solaris South Tower, Singapore 138628, under exclusive license granted by Higher Education Press Limited Company for all media and languages excluding Chinese and throughout the world excluding Mainland China, and with non-exclusive license for electronic versions in Mainland China.

Registered office
John Wiley & Sons Singapore Pte Ltd, 1 Fusionopolis Walk, #07-01 Solaris South Tower, Singapore 138628

For details of our global editorial offices, for customer services and for information about how to apply for permission to reuse the copyright material in this book please see our website at www.wiley.com.

Library of Congress Cataloging-in-Publication Data

Wang, Jie, 1961-
Introduction to network security : theory and practice / Jie Wang, Department of Computer Science at the University of Massachusetts Lowell and Zachary A. Kissel, Department of Computer Science at Merrimack College in North Andover, MA. – Second edition.
pages cm
Includes bibliographical references and index.
ISBN 978-1-118-93948-2 (cloth)
1. Computer networks – Security measures. I. Kissel, Zachary A. II. Title.
TK5105.59.W356 2015
005.8 – dc23

2015021074

A catalogue record for this book is available from the British Library.

ISBN: 9781118939482

Cover Image: Henrik5000/iStockphoto

Typeset in 10/12pt TimesLTStd by SPi Global, Chennai, India

1 2015

Contents

Preface

People today are increasingly relying on public computer networks to conduct business and take care of household needs. However, public networks may be insecure because data stored in networked computers or transmitted through networks can be stolen, modified, or fabricated by malicious users. Thus, it is important to know what security measures are available and how to use them. Network security practices are designed to prevent these potential problems. Originating from meeting the needs of providing data confidentiality over public networks, network security has grown into a major academic discipline in both computer science and computer engineering, and also an important sector in the information industry.

The goal of network security is to give people the liberty of enjoying computer networks without the fear of compromising their rights and interests. Network security accomplishes this goal by providing confidentiality, integrity, nonrepudiation, and availability of useful data that are transmitted in open networks or stored in networked computers.

Network security will remain an active research area for several reasons. Firstly, security measures that are effective today may no longer be effective tomorrow because of advancements and breakthroughs in computing theory, algorithms, and computer technologies. Secondly, after the known security problems are solved, other security loopholes that were previously unknown may at some point be discovered and exploited by attackers. Thirdly, when new applications are developed or new technologies are invented, new security problems may also be created with them. Thus, network security is meant to be a long-lasting scuffle between the offenders and the defenders.

Research and development in network security has mainly followed two lines. One line studies computer cryptography and uses it to devise security protocols. The other line examines loopholes and side effects of the existing network protocols, software, and system configurations. It develops firewalls, intrusion detection systems, anti-malicious-software software, and other countermeasures. Interweaving these two lines together provides the basic building blocks for constructing deep layered defense systems against network security attacks.

This book is intended to provide a balanced treatment of network security along these two lines, with adequate materials and sufficient depth for teaching a one-semester introductory course on network security for graduate and upper-level undergraduate students. It is intended to inspire students to think about network security and prepare them for taking advanced network security courses. This book may also be used as a reference for IT professionals.

This book is a revision and extension of an early textbook written by the first author under the title of "Computer Network Security: Theory and Practice," which was co-published in 2008 by the Higher Education Press and Springer. The book is structured into 10 chapters.

Chapter 1 presents an overview of network security. It discusses network security goals, describes common network attacks, characterizes attackers, and defines a basic network security model.

Chapter 2 presents standard symmetric-key encryption algorithms, including DES, AES, and RC4. It discusses their strength and weaknesses. It also describes common block-cipher modes of operations and a recent block-cipher offset-codebook mode of operations. Finally, it presents key generation algorithms.

Chapter 3 presents standard public-key encryption algorithms and key-exchange algorithms, including Diffie–Hellman key exchange, RSA public-key cryptosystem, and elliptic-curve cryptography. It also discusses how to transmit and manage keys.

Chapter 4 presents secure hash functions and message authentication code algorithms for the purpose of authenticating data, including SHA-512, Whirlpool, SHA-3, cryptographic checksums, and the standard hash message authentication codes. It then discusses birthday attacks on secure hash functions and describes the digital signature standard. It presents a dual signature scheme used for electronic transactions and a blind signature scheme used for producing electronic cash. It concludes with a description of the Bitcoin protocol.

Chapter 5 presents several network security protocols commonly used in practice. It first describes a standard public-key infrastructure for managing public-key certificates. It then presents IPsec, a network-layer security protocol; SSL/TLS, a transport-layer security protocol; and several application-layer security protocols, including PGP and S/MIME for sending secure email messages, Kerberos for authenticating users in local area networks, and SSH for protecting remote logins.

Chapter 6 presents common security protocols for wireless local area networks at the data-link layer, including WEP for providing wired-equivalent privacy, WPA and IEEE 802.11i/WPA2 for providing wireless protected access, and IEEE 802.1X for authenticating wireless users. It then presents the Bluetooth security protocol and the ZigBee security protocol for wireless personal-area networks. Finally, it discusses security issues in wireless mesh networks.

Chapter 7 presents the key security issues involved in the burgeoning area of cloud computing, including a discussion of the multitenancy problem and issues of access control. It then presents advanced topics of searchable encryption for cryptographic cloud storage.

Chapter 8 presents firewall technologies and basic structures, including network-layer packet filtering, transport-layer stateful inspections, transport-layer gateways, application-layer proxies, trusted systems and bastion hosts, screened subnets, and network address translations.

Chapter 9 presents intrusion detection technologies, including intrusion detection system architecture and common intrusion detection methods. It also discusses event signatures, statistical analysis, and data mining methods. Finally, it introduces honeypot technologies.

Chapter 10 describes malicious software, such as viruses, worms, and Trojan horses, and introduces countermeasures. It also covers Web security and discusses mechanisms against denial of service attacks.

Since the publication of the first edition, a number of readers have kindly shared with us their personal experiences in dealing with network security attacks. Some of their stories, after minor editing, are included in the text and the exercise problems.

To get the most out of this book, readers are assumed to have taken undergraduate courses on discrete mathematics, algorithms, data communications, and network programming, or

have equivalent preparations. For convenience, Chapter 3 includes a section reviewing basic concepts and results of number theory used in public-key cryptography. While it does not introduce socket programming, the book contains socket API client–server programming exercises. These exercises are designed for computer science and computer engineering students. Readers who do not wish to do them or simply do not have time to write code may skip them. Doing so would not affect much the learning of materials presented in the book.

Exercise problems for each chapter are divided into discussion problems and homework problems. There are six discussion problems in each chapter, designed to hep stimulate readers to think about the materials presented in that chapter at the conceptual level. These problems are intended to be discussed in class, with the instructor being the moderator. The homework problems are designed to have three levels of difficulty: regular, difficult (designated with *), and challenging (designated with **). This book contains a number of hands-on drills, presented as exercise problems. Readers are encouraged to try them all.

This book is intended to provide a concise and balanced treatment of network security with sufficient depth suitable for teaching a one-semester introductory course on network security. It was written on the basis of what the first author learned and experienced during the last 18 years from teaching these courses and on student feedback accumulated over the years. Powerpoint slides of these lectures can be found at `http://www.cs.uml.edu/~wang/NetSec`. Due to space limitations, some interesting topics and materials are not presented in this book. After all, one book can only accomplish one book's mission. We only hope that this book can achieve its objective. Of course, only you, the reader, can be the judge of it. We will be grateful if you will please offer your comments, suggestions, and corrections to us at wang@cs.uml.edu or kisselz@merrimack.edu.

We have benefited a great deal from numerous discussions over the last 20 years with our academic advisors, colleagues, teaching assistants, as well as current and former students. We are grateful to Sarah Agha, Stephen Bachelder, Yiqi Bai, William Baker, Samip Banker, David Bestor, Robert Betts, Ann Brady, Stephen Brinton, Jeff Brown, William Brown, Matthew Byrne, Robert Carbone, Jason Chan, Guanling Chen, Mark Conway, Michael Court, Andrew Cross, Daniel DaSilva, Paul Downing, Matthew Drozdz, Chunyan Du, Paul Duvall, Adam Elbirt, Zheng Fang, Daniel Finch, Jami Foran, Xinwen Fu, Anthony Gendreau, Weibo Gong, Edgar Goroza, Swati Gupta, Peter Hakewessell, Liwu Hao, Steve Homer, Qiang Hou, Marlon House, Bei Huang, Jared Karro, Christopher Kraft, Fanyu Kong, Lingfa Kong, Zaki Jaber, Ming Jia, Kimberly Johnson, Ken Kleiner, Minghui (Mark) Li, You (Stephanie) Li, Joseph Litman, Benyuan Liu, Yan (Jenny) Liu, Wenjing Lou, Jie Lu, Shan (Ivory) Lu, David Martin, Randy Matos, Laura Mattson, Thomas McCollem, Caterina Mullen, Paul Nelson, Dane Netherton, Michael Niedbala, Gerald Normandin, Kelly O'Donnell, Sunday Ogundijo, Xian Pan, Alexander Pennace, Sandeep Sahu, Subramanian Sathappan, John Savage, Kris Schlatter, Patrick Schrader, Susan Schueller, Liqun (Catherine) Shao, Blake Skinner, Chunyao Song, Adnan Suljevic, Hengky Susanto, Anthony Tiebout, David Thompson, Nathaniel Tuck, John Uhaneh, John Waller, Tao Wang, Brian Werner, Brian Willner, Christopher Woodard, Fang Wu, Jianhui Xie, Jie (Jane) Yang, Zhijun Yu, and Ning Zhong for their comments and feedbacks.

During the writing of the first edition, Jared Karro read the entire draft, Stephen Brinton read Chapters 1–5 and 7–8 (cloud security not included), Guanling Chen read Chapter 6, and Wenjing Lou read Chapters 2 and 6. Their comments have helped improve the quality of the

first edition in many ways, and to them we owe our gratitude. We are grateful to Anthony Gendreau and Adnan Suljevic for pointing out typos in the first edition.

We thank the reviewers for interesting suggestions and Ying Liu at the Higher Education Press for initiating this book project and editing the first edition of the book.

Jie Wang

Zachary A. Kissel

About the Author

Dr. Jie Wang is Professor and Chair of Computer Science at the University of Massachusetts Lowell. He is also Director of the University Center for Internet Security and Forensics Education and Research. He received a Ph.D. degree in Computer Science from Boston University in 1990, an M.S. degree in Computer Science from Zhongshan University in 1985, and a B.S. degree in Computational Mathematics from Zhongshan University in 1982. He has over 23 years of teaching and research experience at the university level and has worked as a network security consultant in the financial industry. He represented the University of Massachusetts system in the education task force of the Advanced Cyber Security Center in New England from 2011 to 2013. His research interests include network security, big data modeling and applications, algorithms and computational optimization, computational complexity theory, and wireless sensor networks. His research has been funded continuously by the National Science Foundation since 1991. His research has also been funded by IBM, Intel, and the Natural Science Foundation of China. He has published over 160 journal and conference papers, six books and four edited books. He is active in professional service, including chairing conference program committees and organizing workshops, editing journals and serving as the editor-in-chief of a book series on mathematical modeling.

Dr. Zachary A. Kissel is Assistant Professor of Computer Science at Merrimack College in North Andover, MA. He received a Ph.D. degree in Computer Science from the University of Massachusetts Lowell in 2013, an M.S. degree in Computer Science from Northeastern University in 2007, and a B.S. degree in Computer Science from Merrimack College in 2005. He has network security industry experience working in a security group at Sun Mircrosystems (later Oracle) where he was responsible for maintaining firewalls and cryptographic libraries. His research interests include cryptography and network security. His work has focused mainly on searchable symmetric encryption and access control for data stored on an untrusted cloud.

1

Network Security Overview

If you know your enemies and know yourself, you will win hundred times in hundred battles. If you know yourself but not your enemies, you will suffer a defeat for every victory won. If you do not know yourself or your enemies, you will always lose.

—Sun Tzu, "The Art of War"

The goal of network security is to give people the freedom to enjoy computer networks without the fear of compromising their rights and interests. Network security therefore needs to guard networked computer systems and protect electronic data that is either stored in networked computers or transmitted in the networks. The Internet, which is built on the IP communication protocols, has become the dominant computer network technology. It interconnects millions of computers and edge networks into one immense network system. The Internet is a public network, where individuals or organizations can easily become subscribers of the Internet service by connecting their own computers and networking devices (e.g., routers and sniffers) to the Internet and paying a small subscription fee.

Because IP is a store-forward switching technology, where data is transmitted using routers controlled by other people, user A can read user B's data that goes through user A's network equipment. Likewise, user A's data transmitted in the Internet may also be read by user B. Hence, any individual or any organization may become an attacker, a target, or both. Even if one does not want to attack other people, it is still possible that one's networked computers may be compromised into becoming an attacking tool. Therefore, to achieve the goal of network security, one must first understand the attackers, what could become their targets, and how these targets might be attacked.

1.1 Mission and Definitions

The tasks of network security are to provide *confidentiality*, *integrity*, *nonrepudiation*, and *availability* of useful *data* that are transmitted in public networks or stored in networked computers.

Introduction to Network Security: Theory and Practice, Second Edition. Jie Wang and Zachary A. Kissel.

The concept of data has a broad sense in the context of network security. Any object that can be processed or executed by computers is data. Thus, source code, executable code, files in various formats, email messages, digital music, digital graphics, and digital video are each considered data. Data should be read, written, or modified only by legitimate users. That is, unauthorized individuals or organizations are not allowed to have access to data.

Just as CPU, RAM, hard disk, and network bandwidth are resources, data is also a resource. Data is sometimes referred to as *information* or *messages*.

Each piece of data has two possible states, namely, the *transmission state* and the *storage state*. Data in the transmission state is simply data in the process of being delivered to a network destination. Data in the storage state is that which is stored in a local computer or in a storage device. Thus, the meanings of data confidentiality and data integrity have the following two aspects:

1. Provide and maintain the confidentiality and integrity of data that is in the transmission state. In this sense, confidentiality means that data during transmission cannot be read by any unauthorized user, and integrity means that data during transmission cannot be modified or fabricated by any unauthorized user.
2. Provide and maintain the confidentiality and integrity of data that is in the storage state. Within this state, confidentiality means that data stored in a local device cannot be read by any unauthorized user through a network, and integrity means that data stored in a local device cannot be modified or fabricated by any unauthorized user through a network.

Data nonrepudiation means that a person who owns the data has no way to convince other people that he or she does not own it.

Data availability means that attackers cannot block legitimate users from using available resources and services of a networked computer. For example, a computer system infected with a virus should be able to detect and disinfect the virus without much delay, and a server hit by denial of service attacks should still be able to provide services to its users.

Unintentional components in protocol specifications, protocol implementations, or other types of software that are exploitable by attackers are often referred to as *loopholes*, *flaws*, or *defects*. They might be an imperfect minor step in a protocol design, an unforeseen side effect of a certain instruction in a program, or a misconfigured setting in a system.

Defense is the guiding principle of network security, but it is a passive defense because before being attacked, the victim has no idea who the attackers are and from which computers in the jungle of the Internet the attackers will launch their attacks. After a victim is attacked, even if the attacker's identity and computer system are known, the victim still cannot launch a direct assault at the attacker, for such actions may be unlawful. What constitutes legal actions against attackers involves a discussion of relevant laws, which is beyond the scope of this book. Therefore, although offense may be the best defense in military operations, this tactic may not apply to network security. Building a deep layered defense system is instead the best possible defense tactic in network security. Within this type of defense system, multiple layers of defense mechanisms are used to resist possible attacks.

Network security is a major part of information security. In addition to network security, information security deals with many other security issues, including security policies, security auditing, security assessment, trusted operating systems, database security, secure

code, emergency response, computer forensics, software forensics, disaster recovery, and security training.

- Security policies are special rules to protect a computer network system against security attacks. For example, security policies may specify what types of data are to be protected, who should be given the access right of read from or write to the data, and how the data should flow from one place to the next.
- Security auditing is a procedure of checking how well the security policies for a particular computer network system are followed. It may be a manual procedure or an automated procedure run by software tools.
- Security assessment is a procedure of determining the security needs of a particular system, measuring the strength and weakness of the existing security policies, and assessing whether the security policies are reasonable and whether security loopholes exist.
- A trusted operating system is an operating system without any security flaws or loopholes in system designs, computing resource management, software implementations, and configurations.
- Database security is a set of security measures specifically devised for database systems, specifying which data fields are accessible by which level of users.
- Secure software is software that contains no security flaws, loopholes, or side effects.
- Intrusion response is a set of actions that should take place when a computer network system is detected being intruded by intruders.
- Cyber forensics studies how to collect information of user activities from computer systems and network communications, providing evidence to indict cyber criminals. Cyber forensics can be further divided into computer forensics and network forensics.
- Disaster recovery is a set of mechanisms to bring a computer system that goes down because of attacks or natural disasters back to a working status.

This book does not cover these issues, but it may touch certain aspects of them.

1.2 Common Attacks and Defense Mechanisms

Common network security attacks can be characterized into a few basic types. Almost every known network security attack is either one of these basic types or a combination of several basic types.

1.2.1 Eavesdropping

Eavesdropping is an old and effective method for stealing private information. In network communications, the eavesdroppers may intercept data from network traffic using a networking device and a packet sniffer. A packet sniffer, or network sniffer, is a program for monitoring incoming network traffic. When connecting a router to the Internet, for example, one can use a packet sniffer to capture all the IP packets going through that router. `TCPdump` and `Wireshark` (formerly known as `Ethereal`) are network sniffers widely used today, which are available as free downloads (see Exercise 1.5).

Using a packet sniffer as an eavesdropping tool, one can intercept IP packets that go through the router he controls. To capture a particular IP packet, however, the eavesdropper must first determine which communication path the IP packet will travel through. Then, he could either try to get control of a certain router on the path or try to insert a new router of his own on the path. This task is more difficult but is not impossible. For example, the eavesdropper may try to compromise a router on the path and install a packet sniffer in it to intercept the IP packets he is after. The eavesdropper may also use an ARP spoofing technique (see Section 1.2.4) to reroute IP packets to his sniffer without compromising a router.

Eavesdropping wireless communications is easier. In this case, the attacker simply needs to place a receiver with the same radio frequency of the wireless network within the communication range of the network.

There is no way to stop eavesdropping in public networks. To counter eavesdropping, the best defense mechanism is to encrypt data. Computer cryptography is developed for this purpose, where the sender encrypts data into an unintelligible form before he transmits it. Data encryption is a major component of computer cryptography. It uses an encryption key in concert with an encryption algorithm, to break the original data into pieces and mix them up in a certain way to make it unintelligible, so that the eavesdropper cannot obtain any useful information out of it. Thus, even if the eavesdropper is able to intercept the encrypted data, he is still not able to obtain the original data without knowing the decryption key. We often refer the original data as *plaintext* data, or simply plaintext, and encrypted data as *ciphertext* data, or simply ciphertext.

Ciphertext data can be converted back to plaintext data using a decryption key along with a decryption algorithm. The encryption key is a string of characters, which is also referred to as *secret key*. In a symmetric-key encryption algorithm, also referred to as conventional encryption, the encryption key and the decryption key are identical. In a public-key encryption algorithm, also known as asymmetric-key encryption, the encryption key and the decryption key are different.

1.2.2 Cryptanalysis

Cryptanalysis is the art and science of finding useful information from ciphertext data without knowing the decryption keys. For example, in a substitution cipher that substitutes plaintext letters with ciphertext letters, if a ciphertext message reveals a certain statistical structure, then one may be able to decipher it. To obtain a statistical structure of the data, one may calculate the frequency of each character in the ciphertext data and compare it against the known statistical frequency of each character in the language used in the plain text. For example, in the English language, the letter "e" has the highest frequency. Thus, in a substitution cipher, the character that has the highest frequency in the ciphertext data is likely to correspond to the plaintext letter "e" (see e.g., Exercise 1.7). This analysis can be further extended to common phrases. Analyzing statistical structures of ciphertext messages was an effective method to break encryptions before the computer era.

Modern encryption algorithms can produce ciphertext without any trace of statistical structure. Therefore, modern cryptanalysis is focused on analyzing encryption algorithms using mathematical techniques and high-performance computers.

The best method against cryptanalysis is to devise encryption algorithms that reveal no statistical structures in ciphertext messages using sophisticated mathematics and longer

encryption keys. Using sophisticated mathematics makes mathematical analysis difficult. Using longer keys makes brute force attacks impractical. In addition to having stronger encryption algorithms, it is equally important to distribute and manage keys safely and to implement encryption algorithms without exploitable loopholes.

1.2.3 Password Pilfering

Computer users need to prove to the system that they are legitimate users. The most widely used authentication mechanism is in the form of user names and user passwords. User names are public information, but user passwords must be kept secret. Only two parties should have knowledge of the password, namely, the user and the underlying computer program (e.g., an operating system or a specific software application). A password is a sequence of letters, digits, or other characters, which is often selected by the user. Legitimate users enter their user names and passwords to prove their legitimacy to the computer program. An unauthorized user may impersonate a legitimate user to "legitimately" log on to a password-protected system or application, if he can get hold of a legitimate user name and password pair. He can then gain all the "legal" rights to transmit, receive, modify, and fabricate data.

Password protection is often the first defense line, and sometimes, it may be the only defense mechanism available in the system. Thus, we must take measures to ensure that user passwords are well protected against larcenies. For this purpose, we will look at several common methods for pilfering user passwords. These methods include *guessing*, *social engineering*, *dictionary attacks*, *side-channel attacks*, and *password sniffing*. *Phishing* attacks and *pharming* attacks have become the most common form of mass social engineering attacks in recent years.

1.2.3.1 Guessing

Guessing is the simplest method to acquire a password illegitimately. The attacker may get lucky if users use short passwords or if they forget to change the default passwords created for them. Also, users have a tendency to use the same passwords.

According to data compiled yearly by SplashData, a password management company, the top 10 most common passwords used by users, listed in decreasing order of popularity, are as follows:

1. `123456`
2. `password`
3. `12345678`
4. `qwerty`
5. `abc123`
6. `123456789`
7. `111111`
8. `1234567`
9. `iloveyou`
10. `adobe123`

If the user chooses a simple password such as these 10 easy ones, then the guesser would indeed have an easy task.

1.2.3.2 Social Engineering

Social engineering is a method of using social skills to pilfer secret information from the victims. For example, attackers may try to impersonate people with authority or organizations of reputation to trick unvigilant users to reveal their user names and user passwords to the attackers. Impersonation may be carried out either in person or in an electronic form. Phishing and pharming are common electronic forms of social engineering attacks in recent years, targeted at a large number of people.

There are other forms of social engineering attacks. For example, attackers may try to collect recycled papers from the recycle bins in a corporation's office building, hoping to find useful login information. Attackers may also make a Web browser pop up a window asking for user login information.

Physical Impersonation

Physical impersonation means that the attacker pretends to be a different person to delude the victim. For example, the following imaginary conversion between the attacker and a receptionist named Betty demonstrates how a social engineering attack might be carried out in person:

> Attacker: (Speaking with an authoritative voice.) "Hello, Betty, this is Nina Hatcher. I am Marketing Manager of the China branch office."
>
> Betty: (Thinking that this woman knew my name, my number, and spoke like a manager, she must be whom she said she was.) "Hello, Nina, what can I do for you?"
>
> Attacker: "Betty, I am attending a meeting in Guangzhou to finalize an important deal with a large corporation in China. To close the deal, I'll need to verify certain technical data produced by your group that I believe is still stored in the computer at your site. This is urgent. I tried to log on to your system today, but for some reason it didn't work. I was able to log on to it yesterday though. Is your computer down? Can you help me out here?"
>
> Betty: "Well, I don't know what happened. But you may try the following $\cdots$" (Thinking that she is doing the company a favor by telling the marketing manager how to get into the system.)

Phishing

Phishing attacks are mass social engineering attacks that take advantage of people with a tendency to trust authorities. The main forms of phishing attacks are disguised email messages or masqueraded Websites. For example, attackers (also called *phishers*) send disguised email messages to people as if these messages were from banks, credit card companies, or other financial institutions that people may pay attention to. People who receive such messages are told that there was a security breach in their accounts, and so they are required to verify their account information for security purposes. They are then directed to a masqueraded Website to enter their user names and passwords (e.g., see Exercise 1.15). The following example is a real phishing message verbatim (The reader may notice a number of grammatical errors and format problems.):

From: UML NEW EMAIL <helpdesk@uml.edu>

To:

Date: Wed, Jul 7, 2010 at 2:28 AM

Subject: Re UNIVERSITY I.T.S UPDATE

Welcome to the university of Massachusetts Lowell New webmail system.

Many of you have given us suggestions about how to make the Umass Lowell webmail better and we have listened. This is our continuing effort to provide you with the best email services and prevent the rate of spam messages received in your inbox folder daily. Consequently all in-active old email accounts will be deleted during the upgrade.

To prevent your account from deletion and or being suspended we recommends all email accounts owner users to upgrade to the new email. Fill in your data in the blank space provided;

```
(Email:________),  (User I.D________), (password________)
(Retype password_____________).
```

The University I.T.S

www.uml.edu

Checked by AVG - Version: 8.5.437 Virus Database: 271.1.12840 - Release

This was a blunt phishing attack, in which the phisher simply asked the recipients to fill in the blanks with their passwords. Other more sophisticated phishing emails may contain a bogus Website as a trap to capture account information entered by the victims. Here, the email and the Website are the baits. The sniffing mechanisms hiding behind the Web page are the hook. Most phishing emails, no matter how well they are put together, would often contain the lines of "Something happened with your account, and you need to go to this page to fix it, or your account will be deleted". In general, any phishing email would contain a link to a bogus Website, called a *phishing site*. Phishing sites may look like the real ones, with the purpose of luring careless users to enter useful login information only to be captured by the phisher.

Even if you do not plan to enter any information on the bogus Website, clicking the link in the phishing email may already compromise your computer, for modern phishing techniques make it possible to embed exploits in a Web page, and the exploits will be activated when you open the Web page.

Users may look at the following three things to detect abnormalities: (1) the "From" address, which may look odd; (2) the URL links the phishers want them to click on, which may be similar to but definitely different from the real site (e.g., a URL that looks like Citicard is in reality not the Citibank's real site); and (3) the look and feel of the Website if the user fails to identify any abnormality during the first two items, for the bogus Website would not be exactly the same as the real site. For example, the color scheme may look different. If you receive an email from a bank or a credit card company telling you that your have a problem with your account and asking you for your user name and password, then most likely it is a phishing email, for banks or credit card companies would never send emails to their customers asking for their account information.

Sometimes, a phishing email may contain a line similar to this: "To be removed from this list click here." Do not click on this link, for it will notify the attacker that the user did read the email and consequently more annoying emails may come.

Antiphishing extensions of Web browsers are emerging technology for detecting and blocking phishing sites. *Email scanners* may also be used to identify phishing emails. However, blocking phishing and not blocking legitimate emails is challenging, even with appropriate email scanners. Thus, users may also want to develop their own tools to detect compromised email accounts and disable them before they can send out phishing emails.

1.2.3.3 Pharming

Pharming attacks use Web technologies to redirect users from the URLs they want to visit to a URL specified by the attacker, including changing DNS setting or the hosts file on the victim's computer, where DNS stands for domain-name service. Attacks that change DNS settings are also referred to as DNS poisoning. If an DNS-poisoning attack is launched from an insecure home router or wireless access point, it is also referred to as a drive-by pharming. Reported by Symantec in 2008, the first drive-by pharming attack was targeted at a Mexican bank.

Similarly to phishing attacks, pharming may also be used to pilfer user passwords. But pharming attacks do not need to set up baiting messages as phishing attacks normally do and hence may disguise themselves better and trap people in more easily.

To counter pharming attacks, it is important for users to make sure that their DNS software and the hosts files have not been compromised and that the URL they are visiting is the right one before doing anything else.

1.2.3.4 Dictionary Attacks

For security reasons, only encrypted passwords, that is, not in their original form, should be stored in a computer system. This prevents attackers from learning the passwords even if they break into the system. In early versions of UNIX and Linux operating systems, for example, the encrypted user passwords of the system are stored in a file named `passwd` under directory `/etc`. This encryption is not a one-to-one encryption. Namely, the encryption algorithm can calculate the ciphertext string of a given password, but the ciphertext string cannot be uniquely decrypted. Such an encryption is also referred to as an *encrypted hash*. In early versions of UNIX and Linux operating systems, user names and the corresponding encrypted user passwords stored in the `passwd` file were ASCII strings that could be read by users. In later versions of UNIX and Linux operating systems, however, the encrypted user passwords of the system are no longer stored this way. Instead, they are stored in a file named `shadow` under directory `/etc`, which is an access-restricted system file.

In the Windows NT/XP operating system, for another example, the user names and the encrypted user passwords are stored in the system's registry in a file named `SAM`. They can be read using special tools, for example, `pwdump`.

Dictionary attacks take advantage of the way some people use dictionary words, names, and dates as passwords. These attacks find user passwords from their encrypted forms. A typical dictionary attack proceeds as follows:

1. Obtain information of user names and the corresponding encrypted passwords. This was done, for example, in early versions of Unix or Linux by getting a copy of the