RSA Security's Official Guide to

CRYPTOGRAPHY

Steve Burnett & Stephen Paine

Disclaimer and Limitation of Liability:
The content of this file is copyrighted material of McGraw-Hill. McGraw-Hill makes no representations or warranties as to the accuracy of any information contained in the McGraw-Hill Material, including any warranties of merchantability or fitness for a particular purpose. In no event shall McGraw-Hill have any liability to any party for special, incidental, tort, or consequential damages arising out of or in connection with the McGraw-Hill Material, even if McGraw-Hill has been advised of the possibility of such damages.

CHAPTER 1

Why Cryptography?

*"According to the affidavit in support of the criminal complaint, the Secret Service began investigating this matter when it learned that there had been unauthorized access to [online brokerage] accounts of several [anonymous company] employees. One [anonymous company] employee told authorities that approximately $285,000 had been drained from his [online brokerage] account when an unknown person was able to access his account by calling the online broker and providing a name and social security number. It was later determined that at least eight [anonymous company] employees had been victimized this past spring, and that these eight had lost a total of $700,000 from their stock accounts . . . [anonymous company] officials revealed that while working in the financial department, [the accomplice] had access to confidential employee information such as social security numbers and home addresses."**

If someone tells you, "I don't need security. I have no secrets, nothing to hide," respond by saying, "OK, let me see your medical files. How about your paycheck, bank statements, investment portfolio, and credit card bills? Will you let me write down your Social Security number,

*Source: U.S. Department of Justice, July 20, 2000 credit card numbers, and bank account numbers? What's the PIN for your ATM, credit card, or phone card? What's your password to log on to the network at work? Where do you keep your spare house key?"

The point is that we all have information we want kept private. Sometimes the reason is simply our natural desire for privacy; we would feel uncomfortable if the whole world knew our medical history or financial details. Another good reason is self-protection—thieves could use some kinds of information to rob us. In other words, the motives for keeping a secret are not automatically nefarious.

Corporations also have secrets—strategy reports, sales forecasts, technical product details, research results, personnel files, and so on. Although dishonest companies might try to hide villainous activities from the public, most firms simply want to hide valuable information from dishonest people. These people may be working for competitors, they might be larcenous employees, or they could be *hackers* and *crackers*: people who break into computer networks to steal information, commit vandalism, disrupt service, or simply to show what they can do.

Security Provided by Computer Operating Systems

In the past, security was simply a matter of locking the door or storing files in a locked filing cabinet or safe. Today, paper is no longer the only medium of choice for housing information. Files are stored in computer databases as well as file cabinets. Hard drives and floppy disks hold many of our secrets. How do you lock a hard drive?

How Operating Systems Work

Before we talk about how computer data is protected, let's take a brief look at how computers get and store information. The usual way to access data on a computer or network is to go through the *operating system* (OS), such as DOS, Windows, Windows 95, Windows NT, MacOS, UNIX, Linux, Solaris, or HP/UX. The OS works like an application, taking input, performing operations based on the input, and returning output. Whereas, for example, a spreadsheet application takes the numbers you type into it, inserts them into cells, and possibly performs calculations such as adding columns, an OS takes your commands in the form of mouse clicks, joysticks, touch screens, or keyboard input-commands such as “show a listing of the files in this directory”—and performs the request, such as printing to the screen a list of files. You can also ask the OS to launch a particular application—say, a text editor. You then tell the text editor to open a file. Behind the scenes, the editor actually asks the OS to find the file and make its contents available to the editor.

Virtually all computers built today include some form of protection courtesy of the OS. Let’s take a look at how such protection works.

Default OS Security: Permissions

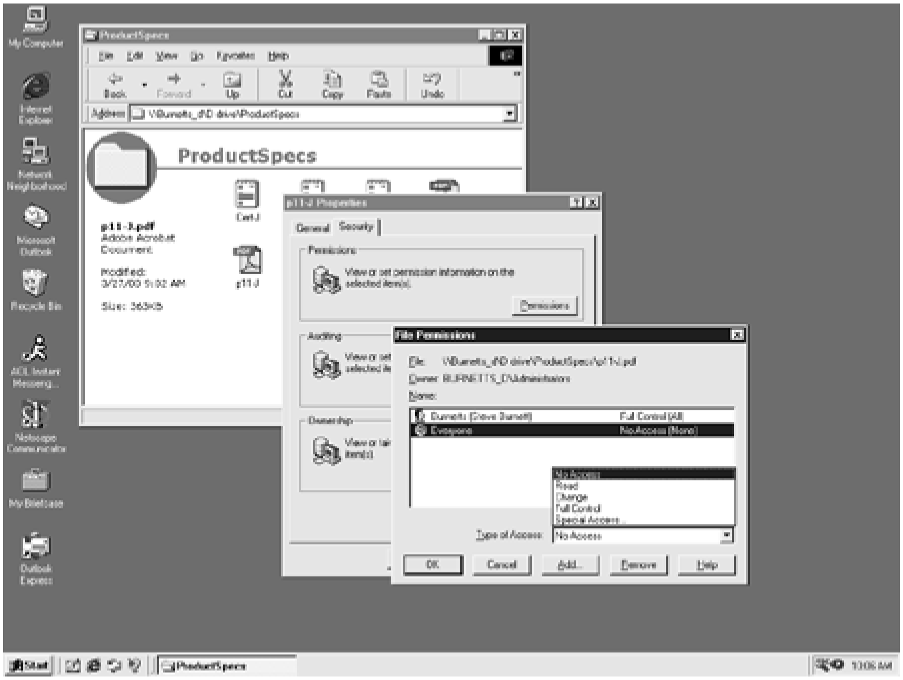
Virtually all operating systems have some built-in *permissions,* which allow only certain people access to the computer (its hard drive, memory, disk space, and network connection). Such access is implemented via a *login* procedure. If the user does not present the appropriate credentials (perhaps a user name and password), the OS will not allow that individual to use the computer. But even after a user is logged in, certain files may still be off-limits. If someone asks to see a file, the OS checks to see whether that requester is on the list of approved users; if not, the OS does not disclose the contents (see Figure 1-1).

Access to most business computers and networks is controlled by someone known as a *superuser* or *system administrator* (often shortened to *sys admin*). This system administrator is the person charged with creating and closing user accounts and maintaining the systems and network. A typical task of this superuser account is to override protections. Someone forgot a password? A file is read-protected (meaning that it cannot be opened and read)? The superuser has permission to circumvent the OS permissions to respond to these problems. (This is where the name “superuser” comes from; this individual can do anything.)

How does the OS know that the person requesting such system overrides is the superuser? The OS grants this access by user name and password. The superuser user name is usually “su” or “root” or “administrator.” Unfortunately, techniques for circumventing these default defenses are widely known.

Figure 1-1

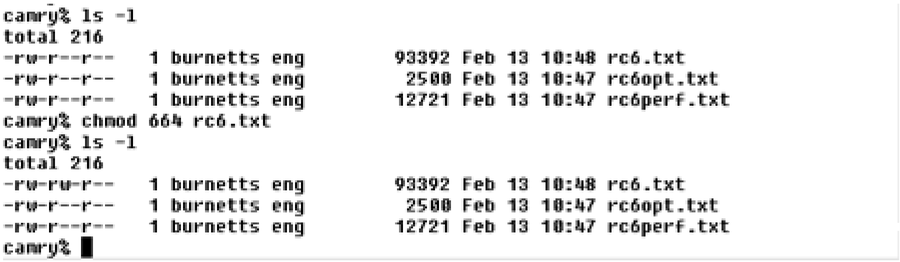
(a) In Windows NT, a file's permission is given in its Properties screen. (b) In UNIX, you type **ls -l** to see a file's permission



File: \\Burnetts_d\D drive\ProductSpecs\p11-1.pdf
Owner: BURNETTS_D\Administrators
Name:

Burnetts (Steve Burnett)	Full Control (All)
Everyone	No Access (None)

Type of Access: No Access



```
camry% ls -l
total 216
-rw-r--r--   1 burnetts eng       93392 Feb 13 10:48 rc6.txt
-rw-r--r--   1 burnetts eng        2500 Feb 13 10:47 rc6opt.txt
-rw-r--r--   1 burnetts eng       12721 Feb 13 10:47 rc6perf.txt
camry% chmod 664 rc6.txt
camry% ls -l
total 216
-rw-rw-r--   1 burnetts eng       93392 Feb 13 10:48 rc6.txt
-rw-r--r--   1 burnetts eng        2500 Feb 13 10:47 rc6opt.txt
-rw-r--r--   1 burnetts eng       12721 Feb 13 10:47 rc6perf.txt
camry% █
```

Attacks on Passwords

Many computers or operating systems come with a preset superuser account and password. In many cases, several passwords are used for various superuser functions. The superuser may have a password to create accounts, a different password to control network functionality, another to conduct or access nightly backups, and so on.

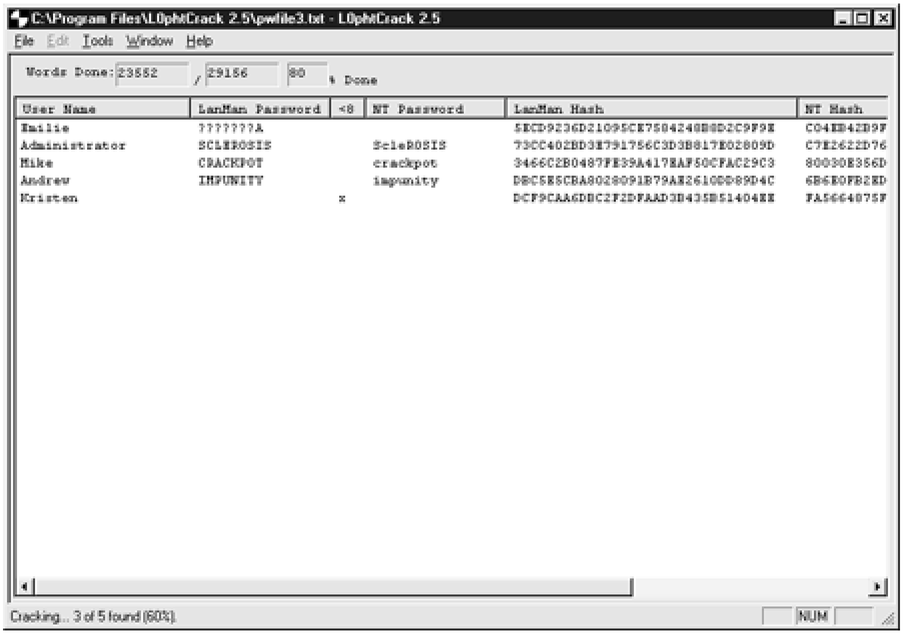
For a cracker, logging on to a system as the superuser is possibly the best way to collect data or do damage. If the superuser has not changed an operating system's preprogrammed passwords, the network is vulnerable to attack. Most crackers know these passwords, and their first attempt to break into a network is simply to try them.

If an attacker cannot log on as the superuser, the next best thing might be to figure out the user name and password of a regular user. It used to be standard practice in most colleges and universities, and in some commercial companies, to assign every student or employee an account with a user name and an initial password—the password being the user name. Everyone was instructed to log on and change the password, but often, hackers and crackers logged on before legitimate users had a chance. In other cases, some people never actually used their accounts. Either way, intruders were able to gain access. This "user name as password" system is still used on many campuses and corporate settings to this day.

If the password of a particular user name is not the user name itself, crackers may try to guess the correct password. Guessing a password might be easy for an insider (such as a fellow employee), who probably knows everyone's user name. It's common for people to use a spouse's name or a birthday as a password. Others write down their passwords, and a quick search of a desk might yield the valuable information. Some systems have guest accounts, with a user name of "guest" and a password of "guest."

But even if the intruder is not very good at guessing passwords, applications are available that automate exhaustive password searches. These applications, called *password cracking* software, are made by a variety of people for various reasons—some legitimate and others not so legitimate. To use one of these tools, the intruder needs access to your computer (network access may be sufficient). Once connected, the hacker simply runs the password cracking application. If the password is weak, within minutes the hacker will have privileged access.

Figure 1-2 shows a popular application known as l0phtCrack. This application is designed to allow systems administrators to test the passwords in use by their users. The idea is that if a sys admin can crack a password, so can crackers.



C:\Program Files\L0phtCrack 2.5\pwfile3.txt - L0phtCrack 2.5

Words Done: 23552 / 29156 80 % Done

User Name	LanMan Password	<8	NT Password	LanMan Hash	NT Hash
Emilie	???????A			5ECD9236D21095CE7584248B8D2C9F9E	C04EB42B9F
Administrator	SCLEROSIS		ScleROSIS	73CC402BD3E791756C3D3B817E02809D	C7E2622D76
Mike	CRACKPOT		crackpot	3466C2B0487FE39A417EAF50CFAC29C3	80030E356D
Andrew	IMPUNITY		impunity	DBC5E5CBA8028091B79AE2610DD89D4C	6B6E0FB2ED
Kristen		x		DCF9CAA6DBC2F2DFAAD3B435B51404EE	FA5664875F

Cracking... 3 of 5 found (60%)

Figure 1-2

l0phtCrack is used to test passwords for vulnerability

Attacks That Bypass Operating Systems

An operating system tags certain files and prevents unapproved people from seeing the contents. Although a cracker or thief might be able to gain access to such files by posing as the superuser or a regular user, another possibility is to ignore the OS altogether and get the contents in some other way.

Data Recovery Attack

One function of a computer's operating system is to help users find and use the specific data or application they want. In this way, an OS works like the index of a book. Just as an index directs you to the specific page where you'll find the piece of information you want out of all the pages in a book, the OS organizes data under a directory file structure and uses file extensions to direct you to the data you want on the hard disk. But as far as the computer is concerned, the data is simply so many electronic bits.

If you don't care what order they're in, it's possible to read those bits as bits and not as files of text or numbers. Human beings can't read bits in this way, but software and hardware devices are available that can scan storage media and read the bits. These tools bypass the OS and grab the raw bits of data, which can then be reconstructed into the original files.

In fact, an entire industry has been built on the concept of reading bits as bits, a process called *data recovery.* When you have a system crash or some kind of physical damage to a hard drive, you can take your computer to a data recovery expert, who often can reconstruct the files on the disk. These companies provide a valuable service, helping to prevent total losses in the event of a natural disaster or computer failure.

Reynolds Data Recovery of Longmont, Colorado, performs data recovery and also sells software that allows you to perform your own recovery (see Figure 1-3). According to the company's advertising, one of its products, Inspector Copier, "does not reference the OS installed on the devices, [and] this allows copies of different systems such as NT, Novell, UNIX, Linux or Windows 2000!"

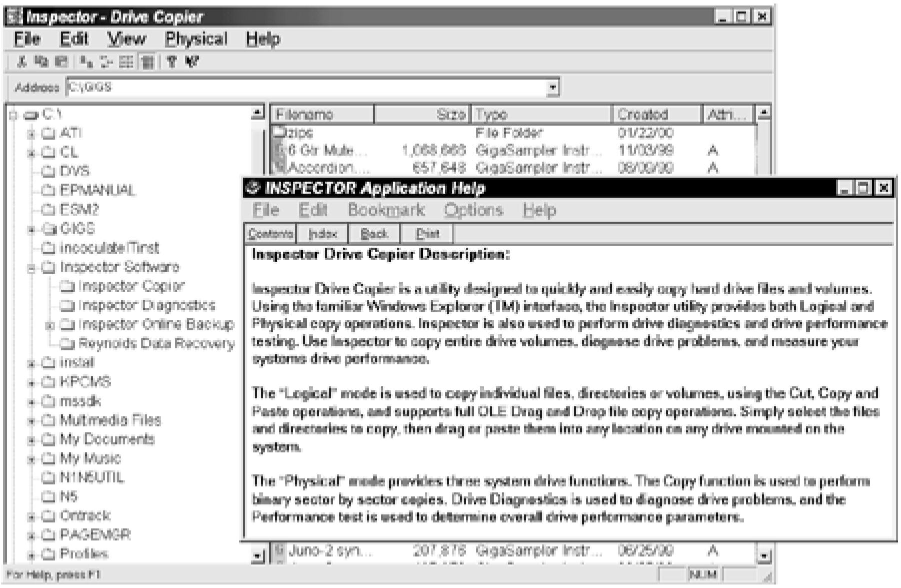


Figure 1-3

Inspector Copier from Reynolds Data Recovery (courtesy of Mark Tessin of Reynolds Data Recovery)

Filename	Size	Type	Created	Attri...
zips		File Folder	01/22/00	
6 Gtr Mute...	1,068,666	GigaSampler Instr...	11/03/99	A
Accordion...	657,648	GigaSampler Instr...	08/09/99	A
Juno-2 syn...	207,876	GigaSampler Instr...	06/25/99	A

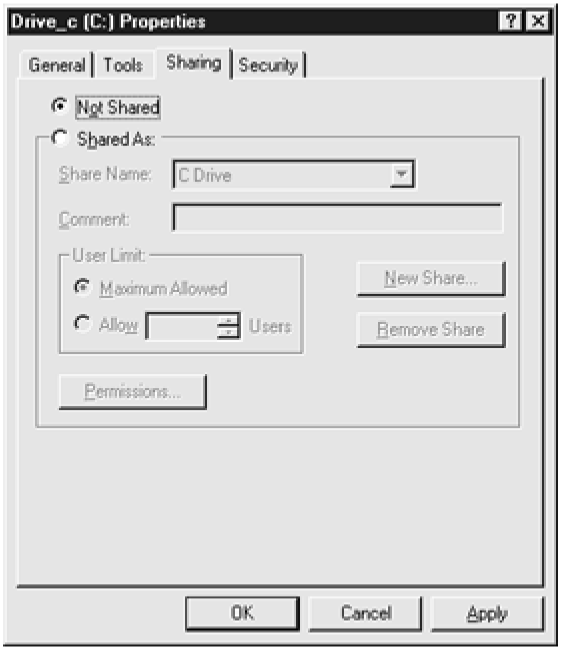
Inspector Drive Copier Description:

Inspector Drive Copier is a utility designed to quickly and easily copy hard drive files and volumes. Using the familiar Windows Explorer (TM) interface, the Inspector utility provides both Logical and Physical copy operations. Inspector is also used to perform drive diagnostics and drive performance testing. Use Inspector to copy entire drive volumes, diagnose drive problems, and measure your systems drive performance.

The "Logical" mode is used to copy individual files, directories or volumes, using the Cut, Copy and Paste operations, and supports full OLE Drag and Drop file copy operations. Simply select the files and directories to copy, then drag or paste them into any location on any drive mounted on the system.

The "Physical" mode provides three system drive functions. The Copy function is used to perform binary sector by sector copies. Drive Diagnostics is used to diagnose drive problems, and the Performance test is used to determine overall drive performance parameters.

But the techniques of data recovery can also be used by attackers to circumvent OS protections. To extend Inspector Copier, Reynolds sells a network backup service that remotely backs up data on hard drives. It uses Inspector Copier to extract the bits so that even if a hard drive is damaged, a clean backup can be made. Although this service can be valuable to many companies, it also means that the data recovery program can be run remotely. Mark Tessin of Reynolds points out that the service can even circumvent Windows NT security. Suppose your PC is connected to a network but you don't want the outside world to see your C: drive. You can set the permissions on your drive so that only you have read or write permission to it (see Figure 1-4). The Reynolds network backup service can circumvent that permission and read the files anyway. This is not to imply that Reynolds Data Recovery will steal your data, only to illustrate that it is possible.



Drive_c (C:) Properties

● Not Shared

○ Shared As:

Share Name: C Drive

Comment:

User Limit:

● Maximum Allowed

○ Allow Users

Figure 1-4

Setting network permissions on a local drive using Windows NT

For serious disk drive failures (such as fire damage), data recovery might be possible only through specialized hardware devices. But an attacker is not trying to steal your data from a damaged drive. Data recovery software is so sophisticated and effective that it's all anyone needs to extract bits from a healthy storage medium.

To ensure the security of your data, you must assume that even though some protections may be sufficient against some opponents, there will likely be someone out there with the resources to mount a successful attack. Only if such an individual never comes after your data are you safe.

Memory Reconstruction Attack

Often, sensitive material is not stored on hard drives but does appear in a computer's memory. For example, when the program you're running allocates some of the computer's memory, the OS tags that area of memory as unavailable, and no one else can use it or see it. When you're finished with that area of memory, though, many operating systems and programs simply "free" it—marking it as available—without overwriting it. This means that anything you put into that memory area, even if you later "deleted" it, is still there. A memory reconstruction attack involves trying to examine all possible areas of memory. The attacker simply allocates the memory you just freed and sees what's left there.

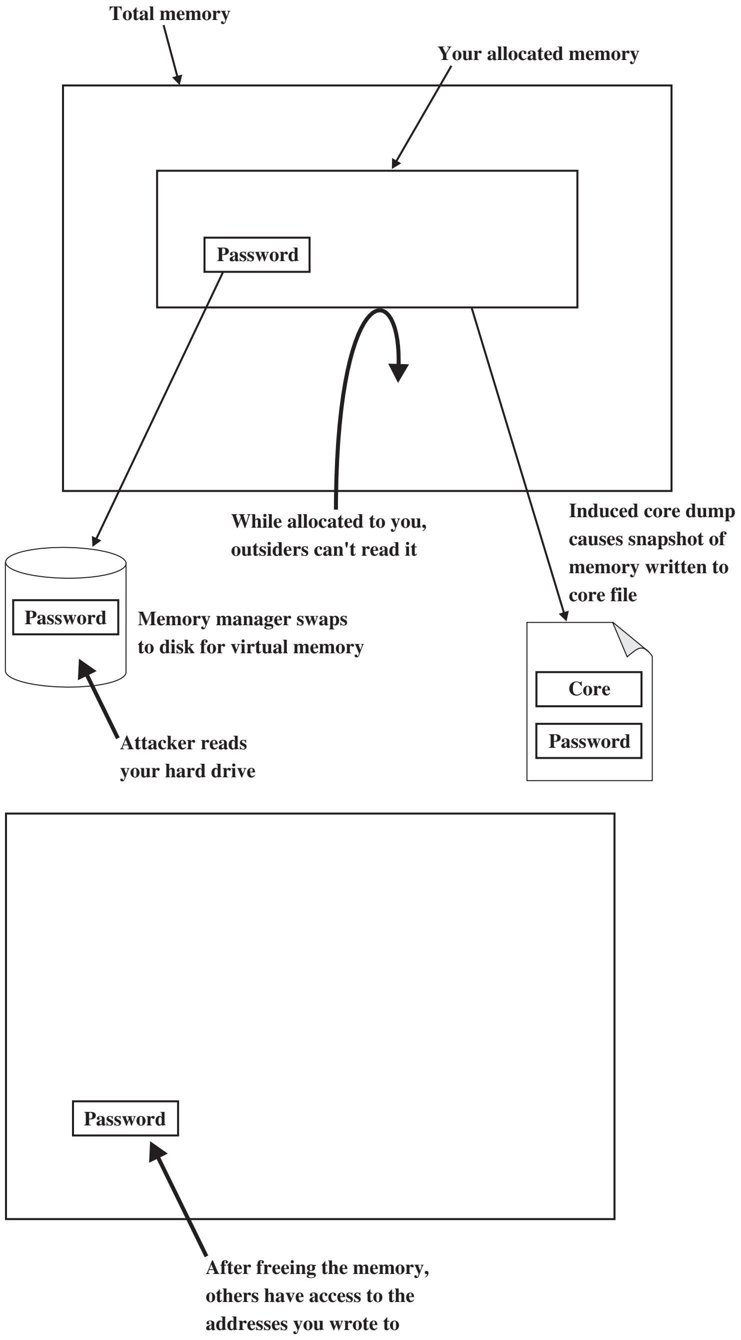
A similar problem is related to what is called "virtual memory." The memory managers in many operating systems use the hard drive as virtual memory, temporarily copying to the hard drive any data from memory that has been allocated but is momentarily not being used. When that information is needed again, the memory manager *swaps* the current virtual memory for the real memory. In August 1997, *The New York Times* published a report about an individual using simple tools to scan his hard drive. In the swap space, he found the password he used for a popular security application.

On UNIX systems, the OS "dumps core" in response to certain system errors. Core dump has become almost synonymous with a program exiting ungracefully. But on UNIX, the core file that results from a core dump is actually a snapshot of memory at the time the error occurred. An attacker who wants to read memory may be able to induce a core dump and peruse the core file.

Figure 1-5 illustrates how memory reconstruction attacks work.

Figure 1-5

Your sensitive material, such a password, is not stored on a hard drive but does appear in memory. An attacker may read the data in memory in the swap space, in a core file, or simply after you free it


Total memory
Your allocated memory
Password
While allocated to you, outsiders can't read it
Induced core dump causes snapshot of memory written to core file
Password
Memory manager swaps to disk for virtual memory
Core
Password
Attacker reads your hard drive
Password
After freeing the memory, others have access to the addresses you wrote to

Added Protection Through Cryptography

For your secrets to be secure, it may be necessary to add protections not provided by your computer system's OS. The built-in protections may be adequate in some cases. If no one ever tries to break into or steal data from a particular computer, its data will be safe. Or if the intruder has not learned how to get around the simple default mechanisms, they're sufficient. But many attackers do have the skills and resources to break various security systems. If you decide to do nothing and hope that no skilled cracker targets your information, you may get lucky, and nothing bad will happen. But most people aren't willing to take that risk.

As you'll learn in the chapters to come, one of the most important tools for protecting data is *cryptography,* any of various methods that are used to turn readable files into gibberish. For example, suppose your sensitive material looks like this:

```
do not believe that the competition can match the new feature set,
yet their support, services, and consulting offerings pose      a
serious threat to our salability. We must invest more money in our
```

Here is what the data looks like when it's encrypted:

```
ú?SdÏ:1/4lYïõ´]Y çmúcA‡[<_b:vH˜_ô UGØ›e´œ_%` ,‹_lo¡`üùØ_"G
ri§õêÌqY_Ë•ùK_æ7ÁFT1≅Ó_ . . . ÀªR8’» ÿÄh . . . o-
2ñ?Í•ÇÕ(tm)ÇvéR]’Î_¬’(r)‹Ñ_UéR`q3/4¥Ü_Ã‡ÁuÉ·¶ _>FômÈÕ6_cêàB1/28#ùh&(G
[gh_!›¶≅Oædtn*´bô1/4jWM1/4B-Â_≅_¬1/4<"-ÏEÿåb{=.AÛH__
```

Even if an attacker obtains the contents of the file, it is gibberish. It does not matter whether or not the OS protections worked. The secret is still secret.

In addition to keeping secrets, cryptography can add security to the process of authenticating people's identity. Because the password method used in almost all commercial operating systems is probably not very strong against a sophisticated (or even an unsophisticated) attacker, it's important to add protection. The cryptographic techniques for providing data secrecy can be adapted to create strong digital identities. If attackers want to pose as someone else, it's not a matter simply of guessing a password. Attackers must also solve an intractable mathematical problem (see Figure 1-6).

Figure 1-6

To pose as Steve Burnett of RSA Security, you'd have to factor this number (see also Chapter 4)

```
111,103,906,294,152,860,689,339,031,055,865,718,
797,834,178,049,634,993,529,562,676,343,628,611,
324,998,912,180,711,483,651,242,218,389,147,835,
598,353,467,199,134,664,870,577,824,583,579,439,
533,042,724,963,790,890,892,988,756,173,576,982,
820,529,088,558,175,928,394,148,986,383,304,407,
218,632,861,415,573,872,050,375,072,884,180,285,
838,244,342,451,974,820,729,610,630,901,524,541,
854,611,490,009,870,503,127
```

The Role of Cryptography in Data Security

In the physical world, security is a fairly simple concept. If the locks on your house's doors and windows are so strong that a thief cannot break in to steal your belongings, the house is secure. For further protection against intruders breaking through the locks, you might have security alarms. Similarly, if someone tries to fraudulently withdraw money from your bank account but the teller asks for identification and does not trust the thief's story, your money is secure. When you sign a contract with another person, the signatures are the legal driving force that impels both parties to honor their word.

In the digital world, security works in a similar way. One concept is *privacy,* meaning that no one can break into files to read your sensitive data (such as medical records) or steal money (by, for example, obtaining credit card numbers or online brokerage account information). Privacy is the lock on the door. Another concept, *data integrity,* refers to a mechanism that tells us when something has been altered. That's the alarm. By applying the practice of *authentication,* we can verify identities. That's comparable to the ID required to withdraw money from a bank account (or conduct a transaction with an online broker). And finally, *nonrepudiation* is a legal driving force that impels people to honor their word.

Cryptography is by no means the only tool needed to ensure data security, nor will it solve all security problems. It is one instrument among many. Moreover, cryptography is not foolproof. All crypto can be broken, and, more importantly, if it's implemented incorrectly, it adds no real security. This book provides an introduction to cryptography with a focus on the proper use of this tool. It is not intended as a complete survey of all there is to know about cryptography. Rather, this book describes the most widely used crypto techniques in the world today.