PART One

In this section of the book, I cover the basics of security engineering technology. The first chapter sets out to define the subject matter by giving an overview of the secure distributed systems found in four environments: a bank, an air force base, a hospital, and the home. The second chapter is on security protocols, which lie at the heart of the subject: they specify how the players in a system—whether people, computers, or other electronic devices—communicate with each other. The third, on passwords and similar mechanisms, looks in more detail at a particularly simple kind of security protocol that is widely used to authenticate people to computers, and provides the foundation on which many secure systems are built.

The next two chapters are on access control and cryptography. Even once a client (be it a phone, a PC, or whatever) has authenticated itself satisfactorily to a server—whether with a password or a more elaborate protocol—we still need mechanisms to control which data it can read or write on the server, and which transactions it can execute. It is simplest to examine these issues first in the context of a single centralized system (access control) before we consider how they can be implemented in a more distributed manner using multiple servers, perhaps in different domains, for which the key enabling technology is cryptography. Cryptography is the art (and science) of codes and ciphers. It is much more than a technical means for keeping messages secret from an eavesdropper. Nowadays it is largely concerned with authenticity and management issues: "taking trust from where it exists to where it's needed" [535].

The final chapter in this part is on distributed systems. Researchers in this field are interested in topics such as concurrency control, fault tolerance, and naming. These take on subtle new meanings when systems must be made resilient against malice as well as against accidental failure. Using old data—replaying old transactions or reusing the credentials of a user who has left some time ago—is a serious problem, as is the multitude of names by which people are known to different systems (email addresses, credit card numbers, subscriber numbers, etc.). Many system failures are due to a lack of appreciation of these issues.

Most of the material in these chapters is standard textbook fare, and the chapters are intended to be pedagogic rather than encyclopaedic, so I have not put in as

many citations as in the rest of the book. I hope, however, that even experts will find some of the case studies of value.

CHAPTER

1

What Is Security Engineering?

Out of the crooked timber of humanity, no straight thing was ever made
—IMMANUEL KANT

The world is never going to be perfect, either on- or offline; so let's not set impossibly high standards for online
—ESTHER DYSON

Security engineering is about building systems to remain dependable in the face of malice, error, or mischance. As a discipline, it focuses on the tools, processes, and methods needed to design, implement, and test complete systems, and to adapt existing systems as their environment evolves.

Security engineering requires cross-disciplinary expertise, ranging from cryptography and computer security through hardware tamper-resistance and formal methods to a knowledge of applied psychology, organizational and audit methods and the law. System engineering skills, from business process analysis through software engineering to evaluation and testing, are also important; but they are not sufficient, as they deal only with error and mischance rather than malice.

Many security systems have critical assurance requirements. Their failure may endanger human life and the environment (as with nuclear safety and control systems), do serious damage to major economic infrastructure (cash machines and other bank systems), endanger personal privacy (medical record systems), undermine the viability of whole business sectors (pay-TV), and facilitate crime (burglar and car alarms). Even the perception that a system is more vulnerable than it really is (as with paying with a credit card over the Internet) can significantly hold up economic development.

The conventional view is that while software engineering is about ensuring that certain things happen ("John can read this file"), security is about ensuring that they don't ("The Chinese government can't read this file"). Reality is much more complex. Security requirements differ greatly from one system to another. One typically needs some combination of user authentication, transaction integrity and accountability, fault-

tolerance, message secrecy, and covertness. But many systems fail because their designers protect the wrong things, or protect the right things but in the wrong way.

In order to see the range of security requirements that systems have to deliver, we will now take a quick look at four application areas: a bank, an air force base, a hospital, and the home. Once we have given some concrete examples of the kind of protection that security engineers are called on to provide, we will be in a position to attempt some definitions.

1.1 Example 1: A Bank

Banks operate a surprisingly large range of security-critical computer systems:

- *The core of a bank's operations is usually a branch bookkeeping system.* This keeps customer account master files plus a number of journals that record the day's transactions. The main threat to this system is the bank's own staff; about one percent of bankers are fired each year, mostly for petty dishonesty (the average theft is only a few thousand dollars). The main defense comes from bookkeeping procedures that have evolved over centuries. For example, each debit against one account must be matched by an equal and opposite credit against another; so money can only be moved within a bank, never created or destroyed. In addition, large transfers of money might need two or three people to authorize them. There are also alarm systems that look for unusual volumes or patterns of transactions, and staff are required to take regular vacations during which they have no access to the bank's premises or systems.
- *The public face of the bank is its automatic teller machines.* Authenticating transactions based on a customer's card and personal identification number—in such a way as to defend against both outside and inside attack—is harder than it looks! There have been many local epidemics of "phantom withdrawals" when villains (or bank staff) have found and exploited loopholes in the system. Automatic teller machines are also interesting as they were the first large-scale commercial use of cryptography, and they helped establish a number of crypto standards.
- *Behind the scenes are a number of high-value messaging systems.* These are used to move large sums of money (whether between local banks or between banks internationally); to trade in securities; to issue letters of credit and guarantees; and so on. An attack on such a system is the dream of the sophisticated white-collar criminal. The defense is a mixture of bookkeeping procedures, access controls, and cryptography.
- *Most bank branches still have a large safe or strongroom, whose burglar alarms are in constant communication with a security company's control center.* Cryptography is used to prevent a robber manipulating the communications and making the alarm appear to say "all's well" when it isn't.
- *Over the last few years, many banks have acquired an Internet presence, with a Web site and facilities for customers to manage their accounts online.* They also issue credit cards that customers use to shop online, and they acquire the resulting transactions from merchants. To protect this business, they use stan-

dard Internet security technology, including the SSL/TLS encryption built into Web browsers, and firewalls to prevent people who hack the Web server from tunneling back into the main bookkeeping systems that lie behind it.

We will look at these applications in later chapters. Banking computer security is important for a number of reasons. Until quite recently, banks were the main non-military market for many computer security products, so they had a disproportionate influence on security standards. Second, even where their technology isn't blessed by an international standard, it is often widely used in other sectors anyway. Burglar alarms originally developed for bank vaults are used everywhere from jewelers' shops to the home; they are even used by supermarkets to detect when freezer cabinets have been sabotaged by shop staff who hope to be given the food that would otherwise spoil.

1.2 Example 2: An Air Force Base

Military systems have also been an important technology driver. They have motivated much of the academic research that governments have funded into computer security in the last 20 years. As with banking, there is not one single application but many:

- *Some of the most sophisticated installations are the electronic warfare systems whose goals include trying to jam enemy radars while preventing the enemy from jamming yours.* This area of information warfare is particularly instructive because for decades, well-funded research labs have been developing sophisticated countermeasures, counter-countermeasures, and so on—with a depth, subtlety, and range of deception strategies that are still not found elsewhere. Their use in battle has given insights that are not available anywhere else. These insights are likely to be valuable now that the service-denial attacks, which are the mainstay of electronic warfare, are starting to be seen on the Net, and now that governments are starting to talk of "information warfare."
- *Military communication systems have some interesting requirements.* It is often not sufficient just to encipher messages: an enemy, who sees traffic encrypted with somebody else's keys may simply locate the transmitter and attack it. *Low-probability-of-intercept* (LPI) radio links are one answer; they use a number of tricks, such as spread-spectrum modulation, that are now being adopted in applications such as copyright marking.
- *Military organizations have some of the biggest systems for logistics and inventory management, and they have a number of special assurance requirements.* For example, one may have a separate stores management system at each different security level: a general system for things like jet fuel and boot polish, plus a second secret system for stores and equipment whose location might give away tactical intentions. (This is very like the business that keeps separate sets of books for its partners and for the tax man, and can cause similar problems for the poor auditor.) There may also be intelligence systems and command systems with even higher protection requirements. The general rule is that sensitive information may not flow down to less-restrictive classifica-

tions. So you can copy a file from a Secret stores system to a Top Secret command system, but not vice versa. The same rule applies to intelligence systems that collect data using wiretaps: information must flow up to the intelligence analyst from the target of investigation, but the target must not know which communications have been intercepted. Managing multiple systems with information flow restrictions is a difficult problem that has inspired a lot of research.

- *The particular problems of protecting nuclear weapons have given rise over the last two generations to a lot of interesting security technology.* These range from electronic authentication systems, which prevent weapons being used without the permission of the national command authority, through seals and alarm systems, to methods of identifying people with a high degree of certainty using biometrics such as iris patterns.

The civilian security engineer can learn a lot from these technologies. For example, many early systems for inserting copyright marks into digital audio and video, which used ideas from spread-spectrum radio, were vulnerable to desynchronization attacks, which are also a problem for some spread-spectrum systems. Another example comes from munitions management, in which a typical system enforces rules such as, "Don't put explosives and detonators in the same truck." Such techniques may be more widely applicable, as in satisfying hygiene rules that forbid raw and cooked meats being handled together.

1.3 Example 3: A Hospital

From food hygiene we move on to healthcare. Hospitals use a number of fairly standard systems for bookkeeping and the like, but also have a number of interesting protection requirements—mostly to do with patient safety and privacy:

- *As Web-based technologies are adopted in hospitals, they present interesting new assurance problems.* For example, as reference books—such as directories of drugs—are moved online, doctors need assurance that life-critical data (such as the figures for dosage per body weight) are exactly as published by the relevant authority, and have not been mangled in some way, whether accidental or deliberate. Many of these safety problems could affect other Web systems in a few years' time. Another example is that as doctors start to access Web pages containing patients' records from home or from laptops in their cars, suitable electronic authentication and encryption tools are starting to be required.
- *Patient record systems should not let all the staff see every patient's record, or privacy violations can be expected.* These systems need to implement rules such as, "nurses can see the records of any patient who has been cared for in their department at any time during the previous 90 days." This can be hard to do with traditional computer security mechanisms, as roles can change (nurses move from one department to another); and there are cross-system dependencies (the patient records system may end up relying on the personnel system for access control decisions, so any failure of the personnel system can have

implications for safety, for privacy, or for both). Applications such as these are inspiring research in role-based access control.

- *Patient records are often anonymized for use in research, but this is difficult to do well.* Simply encrypting patient names is usually not adequate, as an enquiry such as "Show me all records of 59-year-old males who were treated for a broken collarbone on September 15, 1966," would usually be enough to find the record of a politician who was known to have sustained such an injury as a college athlete. But if records cannot be anonymized properly, then much stricter rules will usually have to be followed when handling the data, and this will increase the cost of medical research.
- *New technology can introduce risks that are just not understood.* Hospital administrators understand the need for backup procedures to deal with outages of power, telephone service, and so on, but medical practice is rapidly coming to depend on the Net in ways that are often not documented. For example, individual clinical departments may start using online drug databases; stop keeping adequate paper copies of drug formularies; and never inform the contingency planning team. So attacks that degrade network services (such as viruses and distributed denial-of-service attacks) might have serious consequences for medical practice.

We will look at medical system security in more detail later. This is a much younger field than banking IT or military systems, but as healthcare accounts for a larger proportion of GNP than either of them in all developed countries, and as hospitals are adopting IT at an increasing rate, it looks set to become important.

1.4 Example 4: The Home

You might not think that the typical family operates any secure distributed systems. But consider the following:

- *Many people use some of the systems we've already described.* You may use a Web-based electronic banking system to pay bills; and in a few years you may have encrypted online access to your medical records. Your burglar alarm may send an encrypted "all's well" signal to the security company every few minutes, rather than waking up the neighborhood when something happens.
- Your car may have an electronic immobilizer that sends an encrypted challenge to a radio transponder in the key fob; the transponder has to respond correctly before the car will start. Since all but the most sophisticated thieves now have to tow the car away and fit a new engine controller before they can sell it, this makes theft harder, and reduces your insurance premiums. However, it also increases the number of car-jackings: criminals who want a getaway car are more likely to take one at gunpoint.
- *Early mobile phones were easy for villains to "clone."* Users could suddenly find their bills inflated by hundreds or even thousands of dollars. The current GSM digital mobile phones authenticate themselves to the network by a cryp-

tographic challenge-response protocol similar to the ones used in car-locks and immobilizers.

- Satellite TV set-top boxes decipher movies as long as you keep paying your subscription; DVD players use copy control mechanisms based on cryptography and copyright marking to make it harder to copy disks (or to play them outside a certain geographic area).
- In many countries, households that can't get credit can get prepayment meters for electricity and gas, which they top off using a smartcard or other electronic key which they refill at a local store. Many universities use similar technologies to get students pay for photocopier use, washing machines, and even soft drinks.

The chances are that you already use many systems that enforce some protection policy or other using largely electronic mechanisms. Over the next few decades, the number of such systems is going to increase rapidly. Unfortunately, based on past experience, many of them will be badly designed. The necessary skills are just not spread widely enough.

The aim of this book is to enable you to design such systems better. To do this, an engineer or programmer needs to learn about current systems, how they work, and—at least as important—how they have failed in the past. Civil engineers learn far more from the one bridge that falls down than from the hundred that stay up; exactly the same holds in security engineering.

1.5 Definitions

Many of the terms used in security engineering are straightforward, but some are misleading or even controversial. Though there are more detailed definitions of technical terms in the relevant chapters, which you can find using the index, I point out here where the main problems lie.

The first thing we need to clarify is what we mean by *system*. In practice, this can denote:

1. A product or component, such as a cryptographic protocol, a smartcard, or the hardware of a PC.
2. A collection of the above plus an operating system, communications, and other things that make up an organization's infrastructure.
3. The above plus one or more applications (accounts, payroll, design and so on).
4. Any or all of the above plus IT staff.
5. Any or all of the above plus internal users and management.
6. Any or all of the above plus customers and other external users.
7. Any or all of the above plus the surrounding environment including the media, competitors, regulators, and politicians.

Confusion among these definitions is an extremely fertile source of errors and vulnerabilities. Broadly speaking, the vendor and evaluator communities focus on the first (and occasionally) the second of them, while a business will focus on the sixth (and occasionally the fifth). Ignoring the human components, and thus neglecting usability and liability issues, is one of the primary causes of security failure, so we will generally use definition 6 or 7. When we take a more restrictive view, the meaning should be clear from the context.

The next set of problems comes from lack of clarity about who the players are and what they are trying to prove. In the literature on security and cryptology, it's a convention that principals in security protocols are identified by names chosen with (usually) successive initial letters—much like hurricanes—and so we see lots of statements such as, "Alice authenticates herself to Bob." This makes things much more readable, but often at the expense of precision. Do we mean that Alice proves to Bob that her name actually is Alice, or that she proves she's got a particular credential? Do we mean that the authentication is done by Alice the human being, or by a smartcard or software tool acting as Alice's agent? In that case, are we sure it's Alice, and not perhaps Cherie to whom Alice lent her card, or David who stole her card, or Eve who hacked her PC?

By a *subject* I mean a physical person (human, ET, . . .), in any role including that of an operator, principal, or victim. By a *person*, I mean either a physical person or a legal person such as a company or government.

A *principal* is an entity that participates in a security system. This entity can be a subject, a person, a role, or a piece of equipment, such as a PC, smartcard, or card-reader terminal. A principal can also be a communications channel (which might be a port number or a crypto key, depending on the circumstance). A principal can also be a compound of other principals; examples are a group (Alice or Bob), a conjunction (Alice and Bob acting together), a compound role (Alice acting as Bob's manager), and a delegation (Bob acting for Alice in her absence). Beware that groups and roles are not the same. By a *group* I mean a set of principals, while a *role* is a function assumed by different persons in succession (such as "the officer of the watch on the USS Nimitz" or "the president for the time being of the Icelandic Medical Association"). A principal may be considered at more than one level of abstraction; for example, "Bob acting for Alice in her absence" might mean "Bob's smartcard representing Bob who is acting for Alice in her absence" or even "Bob operating Alice's smartcard in her absence." When I have to consider more detail, I'll be more specific.

The meaning of the word *identity* is controversial. When I am being careful, I will use it to mean a correspondence between the names of two principals signifying that they refer to the same person or equipment. For example, it may be important to know that the Bob in "Alice acting as Bob's manager" is the same as the Bob in "Bob acting as Charlie's manager" and in "Bob as branch manager signing a bank draft jointly with David." Often, the term identity is abused to mean simply "name," an abuse entrenched by such phrases as "user identity" and "citizen's identity card." Where there is no possibility of being ambiguous, I'll sometimes lapse into this vernacular usage in order to avoid pomposity.

The definitions of *trust* and *trustworthy* are often confused. The following example illustrates the difference: if an NSA employee is observed in a toilet stall at Baltimore

Washington International Airport selling key material to a Chinese diplomat, then (assuming his operation was not authorized) he can be described as "trusted but not trustworthy." Hereafter, I'll use the NSA definition that a *trusted* system or component is one whose failure can break the security policy, while a *trustworthy* system or component is one that won't fail.

Beware, though, that there are many alternative definitions of trust. A U.K. military view stresses auditability and fail-secure properties: a trusted systems element is one "whose integrity cannot be assured by external observation of its behavior while in operation." Other definitions often have to do with whether a particular system is approved by authority: a trusted system might be "a system that won't get me fired if it gets hacked on my watch" or even "a system that we can insure." I won't use either of these definitions. When I mean a system that isn't failure-evident, or an approved system, or an insured system, I'll say so.

The definition of *confidentiality* versus *privacy* versus *secrecy* opens another can of worms. These terms clearly overlap; but, equally clearly, they are not exactly the same. If my neighbor cuts down some ivy at our common fence with the result that his kids can look into my garden and tease my dogs, it's not my confidentiality that has been invaded. And the duty to keep quiet about the affairs of a former employer is a duty of confidence, not of privacy.

I'll use these words as follows:

- Secrecy is a technical term that refers to the effect of the mechanisms used to limit the number of principals who can access information, such as cryptography or computer access controls.
- Confidentiality involves an obligation to protect some other person's or organization's secrets if you know them.
- Privacy is the ability and/or right to protect your personal secrets; it extends to the ability and/or right to prevent invasions of your personal space (the exact definition varies quite sharply from one country to another). Privacy can extend to families but not to legal persons such as corporations.

Thus, for example, hospital patients have a right to privacy; in order to uphold this right, the doctors, nurses, and other staff have a duty of confidence toward their patients. The hospital has no right of privacy in respect of its business dealings, but those employees who are privy to them may have a duty of confidence. So, in short, privacy is secrecy for the benefit of the individual, while confidentiality is secrecy for the benefit of the organization.

There is a further complexity in that it's often not sufficient to keep the contents of messages secret. For example, many countries have laws making the treatment of sexually transmitted diseases secret, yet a private eye who could find out that you were exchanging encrypted messages with an STD clinic might well draw the conclusion that you were being treated there. So one may also have to protect metadata such as the source or destination of messages. *Anonymity* can be just as important a factor in privacy (or confidentiality) as secrecy. To make things even more complex, some writers refer to what I've called secrecy as *message content confidentiality*, and to what I've called anonymity as *message source (*or *destination) confidentiality.*

The meanings of *authenticity* and *integrity* can also vary subtly. In the academic literature on security protocols, authenticity means integrity plus freshness: you have established that you are speaking to a genuine principal, not a replay of previous messages. There is a similar idea in banking protocols. In a country whose banking laws state that checks are no longer valid after six months, a seven-month-old uncashed check has integrity (assuming it has not been altered), but is no longer valid. (Bankers would not use the word *authenticity* in this context.) The military usage of authenticity tends to apply to the identity of principals and orders they give, while integrity applies to stored data. Thus, we can talk about the integrity of a database of electronic warfare threats (it has not been corrupted, whether by the other side or by Murphy), but the authenticity of a general's orders (which has an overlap with the academic usage). There are also some strange usages. For example, one can talk about an *authentic copy* of a deceptive order given by the other side's electronic warfare people; here, the authenticity refers to the act of copying and storage. Similarly, a police crime scene officer will talk about preserving the integrity of a cheque that was not authentic but forged, by placing it in an evidence bag.

The last matter I'll clarify here is the terminology that describes what we're trying to achieve. A *vulnerability* is a property of a system or its environment, which, in conjunction with an internal or external *threat*, can lead to a *security failure*, which is a state of affairs contrary to the system's security policy. By *security policy* I mean a succinct statement of a system's protection strategy (for example, "each credit must be matched by an equal and opposite debit, and all transactions over $1,000 must be authorized by two managers"). A *security target* is a more detailed specification, which sets out the means by which a security policy will be implemented in a particular product—encryption and digital signature mechanisms, access controls, audit logs, and so on—and which will be used as the yardstick to evaluate whether the designers and implementers have done a proper job. Between these two levels we may find a *protection profile*, which is like a security target except written in a sufficiently device-independent way to allow comparative evaluations among different products and different versions of the same product. I'll elaborate on security policies, security targets, and protection profiles in Chapter 7 and Chapter 23. In general, the word *protection* will mean a property such as confidentiality or integrity, defined in a sufficiently abstract way for us to reason about it in the context of general systems rather than specific implementations.

Finally, it's worth noting that much of the terminological confusion in security engineering is somewhat political in nature. Security is a terribly overloaded word, and often means quite incompatible things to different people. To a corporation, it might mean the ability to monitor all employees' email and Web browsing activity; to the employees, it might mean being able to use email and the Web without being monitored.

1.6 Summary

I am reminded of a passage from Lewis Carroll:

'When I use a word,' Humpty Dumpty said, in a rather scornful tone, 'it means just what I choose it to mean—neither more nor less.' 'The question is,' said Alice, 'whether you can make words mean so many different things.' 'The question is,' said Humpty Dumpty, 'which is to be master—that's all.'

It is important for the security engineer to develop sensitivity about the different nuances of meaning that common words acquire in different applications, and to be able to formalize what the security policy and target actually are. That may sometimes be inconvenient for clients who wish to get away with something, but, in general, robust security design requires that the protection goals are made explicit.