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

1.1 Introduction

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 economics, applied psychology, organizations 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 (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.

Getting protection right thus depends on several different types of process. You have to figure out what needs protecting, and how to do it. You also need to ensure that the people who will guard the system and maintain it are properly motivated. In the next section, I'll set out a framework for thinking about this. Then, in order to illustrate the range of different things that security systems have to do, I will 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 stuff that security engineers have to understand and build, we will be in a position to attempt some definitions.

1.2 A Framework

Good security engineering requires four things to come together. There's policy: what you're supposed to achieve. There's mechanism: the ciphers, access controls, hardware tamper-resistance and other machinery that you assemble in order to implement the policy. There's assurance: the amount of reliance you can place on each particular mechanism. Finally, there's incentive: the motive that the people guarding and maintaining the system have to do their job properly, and also the motive that the attackers have to try to defeat your policy. All of these interact (see Fig. 1.1).

As an example, let's think of the 9/11 terrorist attacks. The hijackers' success in getting knives through airport security was not a mechanism failure but a policy one; at that time, knives with blades up to three inches were permitted, and the screeners did their task of keeping guns and explosives off as far as we know. Policy has changed since then: first to prohibit all knives, then most weapons (baseball bats are now forbidden but whiskey bottles are OK); it's flip-flopped on many details (butane lighters forbidden then allowed again). Mechanism is weak, because of things like composite knives and explosives that don't contain nitrogen. Assurance is always poor; many tons of harmless passengers' possessions are consigned to the trash each month, while well

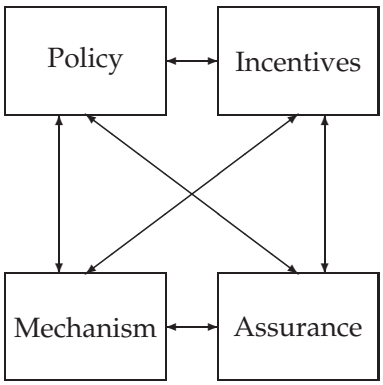

Figure 1.1: Security Engineering Analysis Framework

below half of all the weapons taken through screening (whether accidentially or for test purposes) are picked up.

Serious analysts point out major problems with priorities. For example, the TSA has spent $14.7 billion on aggressive passenger screening, which is fairly ineffective, while $100 m spent on reinforcing cockpit doors would remove most of the risk [1024]. The President of the Airline Pilots Security Alliance notes that most ground staff aren't screened, and almost no care is taken to guard aircraft parked on the ground overnight. As most airliners don't have locks, there's not much to stop a bad guy wheeling steps up to a plane and placing a bomb on board; if he had piloting skills and a bit of chutzpah, he could file a flight plan and make off with it [820]. Yet screening staff and guarding planes are just not a priority.

Why are such poor policy choices made? Quite simply, the incentives on the decision makers favour visible controls over effective ones. The result is what Bruce Schneier calls 'security theatre' — measures designed to produce a feeling of security rather than the reality. Most players also have an incentive to exaggerate the threat from terrorism: politicians to scare up the vote, journalists to sell more papers, companies to sell more equipment, government officials to build their empires, and security academics to get grants. The upshot of all this is that most of the damage done by terrorists to democractic countries comes from the overreaction. Fortunately, electorates figure this out over time. In Britain, where the IRA bombed us intermittently for a generation, the public reaction to the 7/7 bombings was mostly a shrug.

Security engineers have to understand all this; we need to be able to put risks and threats in content, make realistic assessments of what might go wrong, and give our clients good advice. That depends on a wide understanding of what has gone wrong over time with various systems; what sort of attacks have worked, what their consequences were, and how they were stopped (if it was worthwhile to do so). This book is full of case histories. I'll talk about terrorism

specifically in Part III. For now, in order to set the scene, I'll give a few brief examples here of interesting security systems and what they are designed to prevent.

1.3 Example 1 — A Bank

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

1. 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.
2. One 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 epidemics of 'phantom withdrawals' in various countries when local 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.
3. Another public face is the bank's website. Many customers now do more of their routine business, such as bill payments and transfers between savings and checking accounts, online rather than at a branch. Bank websites have come under heavy attack recently from *phishing* — from bogus websites into which customers are invited to enter their passwords. The 'standard' internet security mechanisms designed in the 1990s, such as SSL/TLS, turned out to be ineffective once capable motivated opponents started attacking the customers rather than the bank. Phishing is a fascinating security engineering problem mixing elements from authentication, usability, psychology, operations and economics. I'll discuss it in detail in the next chapter.

4. 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.
5. The bank's branches will often appear to be large, solid and prosperous, giving customers the psychological message that their money is safe. This is theatre rather than reality: the stone facade gives no real protection. If you walk in with a gun, the tellers will give you all the cash you can see; and if you break in at night, you can cut into the safe or strongroom in a couple of minutes with an abrasive wheel. The effective controls these days center on the alarm systems — which are in constant communication with a security company's control center. Cryptography is used to prevent a robber or burglar manipulating the communications and making the alarm appear to say 'all's well' when it isn't.

I'll look at these applications in later chapters. Banking computer security is important: until quite recently, banks were the main non-military market for many computer security products, so they had a disproportionate influence on security standards. Secondly, even where their technology isn't blessed by an international standard, it is often widely used in other sectors anyway.

1.4 Example 2 — A Military 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.

1. 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. As I write, in 2007, a lot of work is being done on adapting jammers to disable improvised explosive devices that make life hazardous for allied troops in Iraq. Electronic warfare has given many valuable insights: issues such as spoofing and service-denial attacks were live there long before bankers and bookmakers started having problems with bad guys targeting their websites.

2. Military communication systems have some interesting requirements. It is often not sufficient to just encipher messages: the enemy, on seeing 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 that are now being adopted in applications such as copyright marking. Covert communications are also important in some privacy applications, such as in defeating the Internet censorship imposed by repressive regimes.
3. Military organizations have some of the biggest systems for logistics and inventory management, which differ from commercial systems in having 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 businessman who keeps separate sets of books for his 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 classifications. 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 which collect data using wiretaps: information must flow up to the intelligence analyst from the target of investigation, but the target must not know which of his communications have been intercepted. Managing multiple systems with information flow restrictions is a hard problem and has inspired a lot of research. Since 9/11, for example, the drive to link up intelligence systems has led people to invent search engines that can index material at multiple levels and show users only the answers they are cleared to know.
4. The particular problems of protecting nuclear weapons have given rise over the last two generations to a lot of interesting security technology, ranging from electronic authentication systems that 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 all this. For example, many early systems for inserting copyright marks into digital audio and video, which used ideas from spread-spectrum radio, were vulnerable to desynchronisation attacks that are also a problem for some spread-spectrum systems. Another example comes from munitions management. There, a typical system enforces rules such as 'Don't put explosives and detonators in the same truck'. Such

techniques can be recycled in food logistics — where hygiene rules forbid raw and cooked meats being handled together.

1.5 Example 3 — A Hospital

From soldiers and food hygiene we move on to healthcare. Hospitals have a number of interesting protection requirements — mostly to do with patient safety and privacy.

1. Patient record systems should not let all the staff see every patient's record, or privacy violations can be expected. They 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 (if the patient records system ends up relying on the personnel system for access control decisions, then the personnel system may just have become critical for safety, for privacy or for both).
2. Patient records are often anonymized for use in research, but this is hard to do well. Simply encrypting patient names is usually not enough as an enquiry such as 'show me all records of 59 year old males who were treated for a broken collarbone on September 15th 1966' would usually be enough to find the record of a politician who was known to have sustained such an injury at college. But if records cannot be anonymized properly, then much stricter rules have to be followed when handling the data, and this increases the cost of medical research.
3. Web-based technologies present interesting new assurance problems in healthcare. For example, as reference books — such as directories of drugs — move 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. Another example is that as doctors start to access patients' records from home or from laptops or even PDAs during house calls, suitable electronic authentication and encryption tools are starting to be required.
4. 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, hospitals in Britain are starting to use online radiology systems: X-rays no longer travel from the X-ray machine to the

operating theatre in an envelope, but via a server in a distant town. So a network failure can stop doctors operating just as much as a power failure. All of a sudden, the Internet turns into a safety-critical system, and denial-of-service attacks might kill people.

We will look at medical system security too 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. In the USA in particular, the HIPAA legislation — which sets minimum standards for privacy — has made the sector a major client of the information security industry.

1.6 Example 4 — The Home

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

1. Many families 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.
2. Your car probably has 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. This makes theft harder and cuts your insurance premiums. But it also increases the number of car thefts from homes, where the house is burgled to get the car keys. The really hard edge is a surge in car-jackings: criminals who want a getaway car may just take one at gunpoint.
3. 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 cryptographic challenge-response protocol similar to the ones used in car door locks and immobilizers.
4. Satellite TV set-top boxes decipher movies so 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). Authentication protocols can now also be used to set up secure communications on home networks (including WiFi, Bluetooth and HomePlug).

5. In many countries, households who can't get credit can get prepayment meters for electricity and gas, which they top up using a smartcard or other electronic key which they refill at a local store. Many universities use similar technologies to get students to pay for photocopier use, washing machines and even soft drinks.
6. Above all, the home provides a haven of physical security and seclusion. Technological progress will impact this in many ways. Advances in locksmithing mean that most common house locks can be defeated easily; does this matter? Research suggests that burglars aren't worried by locks as much as by occupants, so perhaps it doesn't matter much — but then maybe alarms will become more important for keeping intruders at bay when no-one's at home. Electronic intrusion might over time become a bigger issue, as more and more devices start to communicate with central services. The security of your home may come to depend on remote systems over which you have little control.

So you probably already use many systems that are designed to 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. 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 what systems there are, 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.

1.7 Definitions

Many of the terms used in security engineering are straightforward, but some are misleading or even controversial. There are more detailed definitions of technical terms in the relevant chapters, which you can find using the index. In this section, I'll try to point out 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 go to make up an organization's infrastructure;
3. the above plus one or more applications (media player, browser, word processor, accounts / payroll package, 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.

Confusion between the above definitions is a 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). We will come across many examples of systems that were advertised or even certified as secure because the hardware was, but that broke badly when a particular application was run, or when the equipment was used in a way the designers didn't anticipate. Ignoring the human components, and thus neglecting usability issues, is one of the largest causes of security failure. So we will generally use definition 6; when we take a more restrictive view, it 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 will mean a physical person (human, ET, ...), in any role including that of an operator, principal or victim. By a *person*, I will mean either a physical person or a legal person such as a company or government[1].

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 will mean a set of principals, while a *role* is a set of functions 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 considered at more than one level of abstraction: e.g. 'Bob acting

[1]That some persons are not people may seem slightly confusing but it's well established: blame the lawyers.

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 we have to consider more detail, I'll be more specific.

The meaning of the word *identity* is controversial. When we have to be 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, 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) we can describe him as 'trusted but not trustworthy'. Hereafter, we'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 UK military view stresses auditability and fail-secure properties: a trusted systems element is one 'whose integrity cannot be assured by external observation of its behaviour whilst in operation'. Other definitions often have to do with whether a particular system is approved by authority: a trusted system might be 'a system which won't get me fired if it gets hacked on my watch' or even 'a system which we can insure'. I won't use either of these definitions. When we mean a system which 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 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.

The way I'll use these words is as follows.

- *Secrecy* is a technical term which 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 information and extends to the ability and/or right to prevent invasions of your

personal space (the exact definition of which varies quite sharply from one country to another). Privacy can extend to families but not to legal persons such as corporations.

For example, hospital patients have a right to privacy, and in order to uphold this right the doctors, nurses and other staff have a duty of confidence towards 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. 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 protect data, such as the contents of messages; we also have to protect metadata, such as logs of who spoke to whom. For example, many countries have laws making the treatment of sexually transmitted diseases secret, and yet if a private eye could find out that you were exchanging encrypted messages with an STD clinic, he might well draw the conclusion that you were being treated there. (A famous model in Britain recently won a privacy lawsuit against a tabloid newspaper which printed a photograph of her leaving a meeting of Narcotics Anonymous.) So *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 we've called secrecy as *message content confidentiality* and to what we've called anonymity as *message source (or destination) confidentiality*. In general, anonymity is hard. It's difficult to be anonymous on your own; you usually need a crowd to hide in. Also, our legal codes are not designed to support anonymity: it's much easier for the police to get itemized billing information from the phone company, which tells them who called whom, than it is to get an actual wiretap. (And it's often very useful.)

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. We have 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's not been altered) but is no longer valid. The military usage tends to be that authenticity applies 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's 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). However, there are 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 forged check, by placing it in an evidence bag.

The last matter I'll clarify here is the terminology which 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 breach of the system's security policy. By *security policy* I will 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 you 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 later chapters. 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.

1.8 Summary

There is a lot of terminological confusion in security engineering, much of which is due to the element of conflict. 'Security' is a terribly overloaded word, which often means quite incompatible things to different people.

To a corporation, it might mean the ability to monitor all employees' email and web browsing; to the employees, it might mean being able to use email and the web without being monitored. As time goes on, and security mechanisms are used more and more by the people who control a system's design to gain some commercial advantage over the other people who use it, we can expect conflicts, confusion and the deceptive use of language to increase.

One is 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.''*

The security engineer should develop sensitivity to 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.