Computer Science Tripos part 1b — Security

Lecture 1 — Security Applications

Ross J. Anderson

1 Introduction

One might think that just as a building can be made more and more secure by adding more locks, bolts and bars, so a computer system can be made more secure by adding more and more firewalls, password calculators, encryption boxes, and other security products.

This is not the case, and in the first lecture of the course we will start to explain why. We look at a number of applications in order to get some feel for the various combinations of properties such as confidentiality, availability and integrity that a system designer might have to provide. In particular, we will look at bookkeeping systems, transaction processing systems, multilevel secure systems and electronic warfare systems. This will set the stage for our discussion of security mechanisms in later lectures.

When you think about computer security, the first thing that may come to mind is the access control mechanism provided on the laboratory's Unix systems. This identifies you by means of a password, limits you to a certain disk quota, and lets you mark each of your files with an access control list that determines whether it can be read, written or executed and by which users. Almost all time sharing computer systems have systems of this sort; even in a university teaching system where there are no real secrets, they prevent users from deleting each others' files by accident. In fact the one-way password encryption mechanism used in Unix (and many other systems) was developed here at the laboratory in the 1960's during early research into time sharing systems.

However, good engineering proceeds from a requirement to a solution, rather than vice versa. So in this lecture, we are going to talk about examples of requirements for a number of different types of 'secure' system. We will start off by trying to follow the accepted best practice, which follows the 'Waterfall' model and proceeds from a threat model to a security policy and from there to protection mechanisms:

- first, understand the threats against the system. This should wherever possible draw on empirical evidence of actual attacks against similar systems;
- next, write down as concisely as possible what the protection mechanisms are to achieve. This will often be in the form of statements about which

users may access which data. This 'security policy', of which we shall give several examples, may be expressed formally; it will be the baseline for evaluating the system;

- finally, we design the system, using various mechanisms (such as access control lists and cryptographic mechanisms) to enforce the security policy. The strength of mechanisms that we use should be proportional to the threat.

Before we go any further, a word of warning: most University systems are not designed to provide high security, and nothing in this course should be taken as an invitation, or a challenge, to break into our systems (or anybody else's). As we shall discuss later, security holes are all but inevitable in any complex system; they should be reported rather than exploited. Unauthorised access is now an offence under the Computer Misuse Act of 1990, and can also lead to disciplinary action by your college.

2 Bookkeeping Systems

In the commercial world, the threat model is mostly concerned with theft. Theft can take a variety of forms, from the purely opportunist to clever insider frauds; but the experience is that some 60–80% of thefts from the average company are due to insiders. Typical computer crime cases include:

- A bank had a system of suspense accounts, which would be used temporarily if one of the parties to a transaction could not be identified (such as when an account number was entered wrongly on a funds transfer). One of the clerks noticed that no-one investigated a suspense account provided it was cleared within three days. She set up a scheme whereby she would post a debit to a suspense account and an equal credit to her boyfriend's account; after three days, she would raise another debit to pay off the first. In almost two years she netted hundreds of thousands of pounds; she was caught when she could no longer keep track of the growing mountain of bogus transactions.
- A clerk at an education authority wanted to visit relatives in Australia, and in order to get some money she created a fictitious primary school, complete with staff whose salaries were paid into her own bank account. It was only discovered by accident: someone noticed that different records gave the authority different numbers of schools.
- A bank clerk noticed that the branch computer system did not audit address changes. He picked a customer who had a lot of money in her account and got a statement only once a year; he then changed her address to his, issued a new ATM card and personal identification number (PIN), and

changed her address back to its original value. He stole £8,600 from her account, and when she complained she was not believed: the bank maintained that its computer systems were infallible, and so the withdrawals must have been her fault. The matter was only cleared up when the clerk got an attack of conscience and owned up.

In the pre-computer era, the main control against abuses of this kind was double-entry bookkeeping together with regular auditing. Each transaction would be posted to two separate books, as a credit in one and a debit in the other. For example, when a firm is paid £100 by a creditor, the amount is entered as a debit in the accounts receivable (the firm is now owed £100 less) and as a credit in the cash account (the firm now has £100 more cash). In all but the smallest firms, the books will be kept by different clerks, and have to balance at the end of every month (at banks, every day). Thus most frauds will need the collusion of two or more members of staff; and this principle of split responsibility is complemented by audit.

Many computer systems are used for bookkeeping tasks, and implement variations on the double-entry theme. However, all the ledgers may be kept on the same system, and so someone with root access — or with physical access and a debugging tool — may be able to change two or more records so that the balancing controls are bypassed. It may also be possible to evade the balancing controls in various ways — staff may notice bugs in the software and take advantage of them in an opportunistic way.

The standard model of such a system's security policy was introduced in 1987 by David Clark from MIT and David Wilson from Ernst & Whinney. In the Clark-Wilson model, as it is called, some data items are constrained so that they can only be acted on by a certain set of transformation procedures. For example, a well designed banking system will only let a teller update a customer's bank statement if the money comes from somewhere (that is, every credit is matched by an equal debit) and a record of the transaction is securely written on an audit trail which is an append-only file. The transformation procedure consists of all these actions together: it should never be possible for anyone to do only part of a transaction, such as by posting a credit without a debit!

3 Transaction Processing Systems

A growing number of secure computing systems are concerned with transaction processing. These include:

- Networks of automatic teller machines (ATMs) and terminals for electronic funds transfer at the point of sale (eftpos). These were the original retail transaction processing systems: ATMs have been around since 1968, and the world installed base is now somewhere between 400,000 and 500,000 machines. The basic idea is that encryption techniques are used to generate and verify PINs in secure hardware devices located within the ATMs and at bank computer centres. If cryptography can confine the security sensitive part of the application to a trusted computing base, or TCB, and then the rest of the system does not have to be verified and validated to anything like the same extent.

- Telephone cards. These range from prepaid cards for public telephones to the much more sophisticated 'subscriber identity modules', or SIMs, used in GSM digital mobile phones. The SIMs are smartcards which identify the user of a telephone to the network for billing purposes, manage keys for encrypting the conversation, and may even let the subscriber perform banking functions and place bets on horse races. Although only 4 million GSM phones are in use — mostly in Europe — the market is growing at 70% per annum, and 61% of new mobile phone subscribers in the UK now opt for GSM rather than the analogue alternatives.

- Prepayment electricity and gas meters. Britain has over a million of these, and supplying them to developing countries is a significant industry; South Africa plans to install over two million of them by 1999 as part of the development programme, and the system should eventually extend to all of subsaharan Africa.

- Road toll and parking garage tokens. Some countries issue these tokens to all their motorists; others may use multipurpose tokens, and there is a scheme being piloted in Germany to enable road tolls to be paid using car telephones.

- Lottery ticket terminals. The national lottery uses encryption to ensure that vendors cannot manufacture valid tickets after the draw or otherwise manipulate the system.

- Postal franking machines. Thanks to cryptography, these can now be replenished remotely: you call and buy a 'magic number' which enables your machine to dispense a certain amount of postage.

In many ways the typical application is the ATM, and the security mechanisms used in electricity meters, road tolls and the like have been copied more or less directly from the ATM world. We append an article which discusses ATM security and in particular the things that tend to go wrong. This threat model is also relevant for other transaction processing systems.

4 Electronic Warfare

The encryption techniques used in ATMs were inspired by military systems, and in particular by the identify-friend-or-foe system which warplanes carry to avoid being shot down by their own side's air defences. So we should next take a look at some of the military applications.

Military and diplomatic organisations first developed cryptology, and gave us the subject's vocabulary: it was William Friedman, America's star codebreaker in both world wars, who defined 'cryptography' as the techniques for devising code and cipher systems, 'cryptanalysis' as the art of breaking them, and 'cryptology' as the science which includes both of these.

However, simply sending messages backwards and forwards encrypted under some electronic, mechanical or even manual cipher system is only a small part of the story nowadays. Another pioneer, Butler Lampson (who discovered covert channels) once remarked that anyone who claims that a problem can be solved by cryptology doesn't understand his problem and doesn't understand cryptology.

Military systems also use cryptographic techniques in a wide variety of applications, including:

- Frequency agile radar. If your opponent can predict your next radar pulse, then he can send a false echo ahead of the real one and seem to be closer to you than he actually is. If, on the other hand, you make your transmitter unpredictable — by varying the frequency, the timing or the modulation of the pulses — he will not be able to do this, and any jamming signals should arrive at your receiver after the real returns. This makes jamming significantly harder.

- Spread spectrum radio. If you launch a missile at an enemy aircraft, it may depend for all or most of its flight on guidance from your own radar. As soon as the target detects your launch, he may try to jam the guidance signal. The standard countermeasure is to modulate this signal with a high speed pseudorandom sequence and thus spread its spectrum out over many megahertz. This can give you a substantial jamming margin; if the enemy cannot predict the sequence, he will have to jam a broad range of frequencies, which takes a lot of energy.

- Low-probability-of-intercept systems. Here the goal is to avoid even the existence of radio traffic being detected; otherwise the enemy might use radio direction finding to locate you and then call down artillery fire on your position. The techniques used range from spread spectrum to meteor scatter.

- Identify friend or foe. Since the 1970's, the majority of air combat kills have been achieved with missiles rather than cannon, and enemy aircraft may be engaged at ranges of 20 miles or more. Visual identification is not possible, so how do you know who to shoot? In early days, returning bomber pilots would identify themselves to radar operators by flying in prearranged patterns, but these can be observed by enemy bombers as well. The solution is to send a random challenge in the primary radar pulse, which friendly aircraft can encrypt and return.

So what sort of properties should a military encryption system possess? The answer depends on the application. To support a frequency agile or spread spectrum system, it is sufficient to have a means of generating a stream of bits which appears random to anybody who does not know the key. A pseudorandom generator, or keystream generator as it is also called, can also be used to encrypt data, by exclusive-or'ing it bit by bit with the plaintext to get ciphertext. Such an arrangement is known as a stream cipher.

However, it is rather hard to implement IFF by just getting aircraft to broadcast a keystream. Firstly, there would be synchronisation difficulties; and secondly, a clever enemy might detect the keystream being transmitted by one of your aircraft over his territory, and send it to one of his bombers which retransmits it while penetrating your air defences. For this reason, block ciphers were developed which encrypt a whole string of data at a time, and which can be used to support a challenge-response protocol. We will discuss both stream and block ciphers in more detail in the third lecture.

5 Multilevel Secure Systems

Military organisations possess many information systems in which it is important to keep certain information secret. A good example is an air force stores system; the number and location of an army's bootlaces is not in general of great interest to an enemy, but the location of nuclear warheads is.

The NATO countries have a system of classifying data at one of a series of levels, which in Britain run unclassified — restricted — confidential — secret — top secret. The ideas behind these levels, and examples of data which might be classified at them, are as follows:

Restricted: information with little known military value but which is protected, e.g., because of commercial confidentiality. Typical examples might be purchase orders for, and the location of, stores of food.

Confidential: information whose compromise would 'cause serious damage to ... operational effectiveness', or give rise to a diplomatic protest or significant financial loss, or prejudice individual security or liberty. An example might be government standards for erasing magnetic media and for shielding rooms and equipment against electronic eavesdropping.

Secret: information whose compromise would threaten life directly, or seriously damage relations with friendly governments, or do significant damage to the economy. Examples range from cabinet minutes through the performance data and test standards for weapons systems to the location of nuclear materials and tactical stores.

Top Secret: information whose compromise would lead directly to widespread loss of life, threaten directly the internal stability of the UK or friendly countries, cause severe long-term damage to the economy, or do 'exceptionally grave damage to ... intelligence operations'. Examples range from the identities of UK agents abroad and decrypts of foreign government signals, to the official assessments of the threat posed to UK government signals and computer data by the intelligence agencies of other countries.

The principle underlying the processing of classified data is that information always flows up. A user cleared to Secret can see information classified as Secret or Confidential, but should see nothing classified as Top Secret. This property has to be enforced in computer systems too, and in the old days separate systems were maintained for each level: moving from a Secret file to a Confidential one meant logging off, leaving the room, passing a guard post, and logging on to a different system. Of course, this turned out to be impractical, and a lot of work has been done (mostly in the USA) on building 'multilevel secure systems'.

We will discuss these in detail in the part 2 security course, but the basic idea is to formalise and enforce the property that processes can read down and write up, but never vice versa. The enforcement is usually done by a piece of software called a reference monitor, which supervises all operating system calls and compares the classification and clearance levels to decide whether the call can be serviced or not. An operating system like this causes a number of practical problems: applications often need to be rewritten or modified to cope with it, and for various reasons the protection that can be given is never absolute. All practical systems leak in one way or another, and we will discuss why this is so later.

6 Other Systems

Finally, there are a number of other commercial applications that have interesting properties. They include:

- Satellite TV operators often encrypt their signals, and have sold tens of millions of decoders worldwide — Sky TV has fielded over five million in Britain alone. These decoders are one nonmilitary application of cryptography which has attracted sophisticated and sustained technical attacks.
- Burglar alarms. Alarm systems for high value sites must provide protection against attacks on their signaling systems: one does not want a villain to be able to attach a black box to a bank's telephone line which would send a bogus signal saying 'all's well' to the security company's control room while he was breaking in. There are a number of ways of doing this, including armoured, CO_2-pressurised cables, but the most economic (and strongest) methods all involve cryptography.
- Tachographs. The European commission wants the current system for monitoring transport drivers' hours and speed to be replaced with a smart-

card based system on the grounds that this might be harder to tamper with.

- Computer access tokens. A typical product looks like an ordinary pocket calculator, but its real function is to calculate a new password for you every time you log on. This prevents a number of password guessing and network monitoring attacks.

- Access control products for distributed systems, such as Kerberos, allow a user to log on to a client system using a password that is verified at a security server. This generates a ticket that can be presented to another security server which in turn controls access to a system resource. The idea is to provide scalable security where users and resources may be in different administrative domains. We will discuss Kerberos in more detail later.

- Copyright marking schemes. Encrypted copyright notices and serial numbers may be embedded in digital images, program code and other electronically distributed objects, in order to prove ownership and to trace customers who break licence agreements and resell copies of products.

7 Security Engineering

I hope the discussion so far has led to a realisation that security is a system property, and not something that can easily be retrofitted to a poorly designed system by adding 'security' components. Indeed, security is such an overloaded word that I prefer to minimise use of it and talk instead about 'integrity', 'confidentiality', 'non-repudiation', availability', 'covertness' and so on.

Most systems need a guarantee of some property of this kind, even in the presence of malicious inputs. Delivering such a guarantee is the task of security engineering, and the goal of this course (together with the part 2 security course) is to help you develop a feel for this.

In the part 1a software engineering lectures, I discussed 'critical systems' as systems some aspect of whose performance had to be guaranteed, and focussed on those aspects of systems whose malfunction could lead to loss of life. But almost all that I said about safety critical systems applies to security as well — and we can get off to a good start by considering security engineering in the same way.

Just as when building a safety critical system, we have to start off with a clear understanding of the hazards. Who are the attackers? Are they largely insiders, as with a bank bookkeeping system, or largely outsiders, as when we are trying to prevent jamming of an air defence radar? Will the 'owner' of the system be the attacker and have unsupervised access to it, as is the case with pay-TV decoders and copyright marking schemes, or will the access be subject to periodic supervision, as with prepayment gas meters and lorry tachographs? At the other extreme, access might be restricted to trusted personnel, as with air force communication systems, but even when this appears to be the case there may be untrusted players in the system (not all banks connected to the worldwide ATM system are trustowrthy).

As with safety-critical systems, getting the requirements right is usually the hard part, and the best way to so this is usually to spend days discussing previous security failures in as much detail as possible with knowledgeable people. Even then one must expect the protection to evolve as attackers change their modus operandi in response to new protective mechanisms, and there should be mechanisms for monitoring and regular review feeding through into the system update or redevelopment process.

In an ideal world, the requirements would be crystallised as a security policy, of which the Clark-Wilson model of commercial system security and the Bell LaPadula model of military multilevel security are the two classic examples. Unfortunately, in many systems, the necessary requirements analysis is skimped and the security policy (if there is one at all) is just a few pages of platitudes. Sometimes there are nefarious reasons for this: the system may be designed to transfer power and/or liability in ways about which its owners do not wish to be too explicit. Often there is no malice but merely indifference — especially where the costs of security failures accrue to some other party.

Protection priorities also change over time, or in the light of experience. A fire alarm with weakly protected communications is adapted as a burglar alarm, whose communications are more likely to be attacked; an ATM system is developed in a country where consumers have few rights and so the resolution of disputes between bank and customer is not a priority, and then exported to a country where the onus of proof lies on the bank; a tachograph designed by engineers with inadequate input from policemen tries to prevent technical fraud but remains vulnerable to much more common procedural frauds; intelligence databases carefully ensure that information only flows upwards from 'Secret' to 'Top Secret' but allow Aldrich Ames — the traitor with the 'Top Secret' clearance — to see everything.

Real systems also involve techniques that some people consider to be outside the scope of computer science, ranging from personnel screening through insurance to security printing. A security engineer needs to understand these too; as with safety critical systems, failures often occur at the interfaces (especially the human interfaces) and it is difficult to design good systems without understanding them. this will often require a prototype to be fielded and subjected to real attack.

For all these reasons, the 'Waterfall' approach to security is not always possible and iterative methodologies such as the 'Spiral' and evolutionary approaches become essential. Unfortunately, the security literature has largely ignored the lessons of software engineering in this respect.

For the next three lectures, we will avoid this debate by assuming that we have somehow arrived at an idea of what we are trying to do. We will look at the main technical mechanisms available to the security engineer, including techniques specific to a single machine, such as access control lists, and techniques for transporting trust from one machine to another, such as cryptography.

8 Problems

1. Suppose you are computerising Britain's medical records, and building a distributed database of all GP and hospital records, as well as all drugs prescribed. What would the main security requirements be?

2. Suppose that a census system lets you enquire about the average value of any personal datum — such as family earnings — for each parish. Would this help you to find out Richard Branson's salary? If you ran the Office of National Statistics, what precautions would you take?

3. Sketch a security policy for a national system of electronic tachographs, the function of which is to detect drivers exceeding their legal hours and also record a detailed vehicle speed history for the purpose of investigating accidents, toxic waste dumping and drug trafficking. How would the trust model be altered by changing the primary input from the vehicle's indicated road speed to a GPS receiver signal or to sensors embedded in the road?

4. Describe some of the threats posed by the battlefield capture of an air defence missile battery. If you were designing its control system, what precautions would you take?

5. We mentioned above that a military system has the security policy that processes can only read down and write up. However, Unix does not have two file attributes, but three: read, write and execute. Is there any security policy which one could impose on 'execute' which might make it harder for computer viruses to spread? Would this policy be practical?

6. Many texbooks on security consider the subject to be encompassed by confidentiality, integrity and availability. The US Department of Defence adds a fourth property, authenticity, that relates to identifying users rather than just data (the context is that discussed in exercise 3 above). Are these properties adequate? How would you use them to express copyright marking schemes, burglar alarm signalling, jam resistant radar and covert communications?