6.4 Naming

Naming is a minor if troublesome aspect of ordinary distributed systems, but it becomes surprisingly hard in security engineering. The topical example in the 1990s was the problem of what names to put on public key certificates. A certificate that says simply 'the person named Ross Anderson is allowed to administer machine X' is little use. Before the arrival of Internet search engines, I was the only Ross Anderson I knew of; now I know of dozens of us. I am also known by different names to dozens of different systems. Names exist in contexts, and naming the principals in secure systems is becoming ever more important and difficult.

Engineers observed then that using more names than you need to causes unnecessary complexity. For example, A certificate that simply says 'the bearer of this certificate is allowed to administer machine X' is a straightforward bearer token, which we know how to deal with; but once my name is involved, then presumably I have to present some kind of ID to prove who I am, and the system acquires a further dependency. Worse, if my ID is compromised the consequences could be extremely far-reaching.

Since 9/11 the terms of this debate have shifted somewhat, as many governments have rushed to issue their citizens with ID cards. In order to justify the expense and hassle, pressure is often put on commercial system operators to place some reliance on government-issue ID where this was previously not thought necessary. In the UK, for example, you can no longer board a domestic flight using just the credit card with which you bought the ticket, and you have to produce a passport or driving license to cash a check or order a bank transfer for more than £1000. Such measures cause inconvenience and introduce new failure modes into all sorts of systems.

No doubt this identity fixation will abate in time, as governments find other things to scare their citizens with. However there is a second reason that the world is moving towards larger, flatter name spaces: the move from barcodes (which code a particular product) to RFID tags (which contain a 128-bit unique identifier that code a particular item.) This has ramifications well beyond naming — into issues such as the interaction between product security, supply-chain security and competition policy.

For now, it's useful to go through what a generation of computer science researchers have learned about naming in distributed systems.

6.4.1 The Distributed Systems View of Naming

During the last quarter of the twentieth century, the distributed systems research community ran up against many naming problems. The basic algorithm used to bind names to addresses is known as *rendezvous*: the principal

exporting a name advertises it somewhere, and the principal seeking to import and use it searches for it. Obvious examples include phone books, and directories in file systems.

However, the distributed systems community soon realised that naming can get fiendishly complex, and the lessons learned are set out in a classic article by Needham [958]. I'll summarize the main points, and look at which of them apply to secure systems.

1. *The function of names is to facilitate sharing*. This continues to hold: my bank account number exists in order to provide a convenient way of sharing the information that I deposited money last week with the teller from whom I am trying to withdraw money this week. In general, names are needed when the data to be shared is changeable. If I only ever wished to withdraw exactly the same sum as I'd deposited, a bearer deposit certificate would be fine. Conversely, names need not be shared — or linked — where data will not be; there is no need to link my bank account number to my telephone number unless I am going to pay my phone bill from the account.
2. *The naming information may not all be in one place, and so resolving names brings all the general problems of a distributed system*. This holds with a vengeance. A link between a bank account and a phone number assumes both of them will remain stable. So each system relies on the other, and an attack on one can affect the other. In the days when electronic banking was dial-up rather than web based, a bank which identified its customers using calling line ID was vulnerable to attacks on telephone systems (such as tapping into the distribution frame in an apartment block, hacking a phone company computer, or bribing a phone company employee). Nowadays some banks are using two-channel authorization to combat phishing — if you order a payment online you get a text message on your mobile phone saying 'if you want to pay $X to account Y, please enter the following four digit code into your browser'. This is a bit tougher, as mobile phone traffic is encrypted — but one weak link to watch for is the binding between the customer and his phone number. If you let just anyone notify you of a customer phone number change, you'll be in trouble.
3. *It is bad to assume that only so many names will be needed*. The shortage of IP addresses, which motivated the development of IP version 6 (IPv6), is well enough discussed. What is less well known is that the most expensive upgrade which the credit card industry ever had to make was not Y2K remediation, but the move from thirteen digit credit card numbers to sixteen. Issuers originally assumed that 13 digits would be enough, but the system ended up with tens of thousands of banks (many with dozens of products) so a six digit *bank identification number*

(BIN number) was needed. Some card issuers have millions of customers, so a nine digit account number is the norm. And there's also a *check digit* (a one-digit linear combination of the other digits which is appended to detect errors).

4. *Global names buy you less than you think.* For example, the 128-bit in IPv6 can in theory enable every object in the universe to have a unique name. However, for us to do business, a local name at my end must be resolved into this unique name and back into a local name at your end. Invoking a unique name in the middle may not buy us anything; it may even get in the way if the unique naming service takes time, costs money, or occasionally fails (as it surely will). In fact, the name service itself will usually have to be a distributed system, of the same scale (and security level) as the system we're trying to protect. So we can expect no silver bullets from this quarter. One reason the banking industry was wary of initiatives such as SET that would have given each customer a public key certificate on a key with which they could sign payment instructions was that banks already have perfectly good names for their customers (account numbers). Adding an extra name has the potential to add extra costs and failure modes.
5. *Names imply commitments, so keep the scheme flexible enough to cope with organizational changes*. This sound principle was ignored in the design of a UK government's key management system for secure email [76]. There, principals' private keys are generated by encrypting their names under departmental master keys. So the frequent reorganizations meant that the security infrastructure would have to be rebuilt each time — and that money would have had to be spent solving many secondary problems such as how people would access old material.
6. *Names may double as access tickets, or capabilities*. We have already seen a number of examples of this in Chapters 2 and 3. In general, it's a bad idea to assume that today's name won't be tomorrow's password or capability — remember the Utrecht fraud we discussed in section 3.5. (This is one of the arguments for making all names public keys — 'keys speak in cyberspace' in Carl Ellison's phrase — but we've already noted the difficulties of linking keys with names.)

I've given a number of examples of how things go wrong when a name starts being used as a password. But sometimes the roles of name and password are ambiguous. In order to get entry to a car park I used to use at the university, I had to speak my surname and parking badge number into a microphone at the barrier. So if I say, 'Anderson, 123', which of these is the password? In fact it was 'Anderson', as anyone can walk through the car park and note down valid badge

numbers from the parking permits on the car windscreens. Another example, from medical informatics, is a large database of medical records kept by the UK government for research, where the name of the patient has been replaced by their postcode and date of birth. Yet access to many medical services requires the patient to give just these two items to the receptionist to prove who they are. I will have more to say on this later.

7. *Things are made much simpler if an incorrect name is obvious*. In standard distributed systems, this enables us to take a liberal attitude to cacheing. In payment systems, credit card numbers may be accepted while a terminal is offline so long as the credit card number appears valid (i.e., the last digit is a proper check digit of the first fifteen) and it is not on the hot card list. Certificates provide a higher-quality implementation of the same basic concept.

   It's important where the name is checked. The credit card check digit algorithm is deployed at the point of sale, so it is public. A further check — the *card verification value* on the magnetic strip — is computed with secret keys but can be checked at the issuing bank, the acquiring bank or even at a network switch (if one trusts these third parties with the keys). This is more expensive, and still vulnerable to network outages.

8. *Consistency is hard, and is often fudged. If directories are replicated, then you may find yourself unable to read, or to write, depending on whether too many or too few directories are available*. Naming consistency causes problems for e-commerce in a number of ways, of which perhaps the most notorious is the bar code system. Although this is simple enough in theory — with a unique numerical code for each product — in practice it can be a nightmare, as different manufacturers, distributors and retailers attach quite different descriptions to the bar codes in their databases. Thus a search for products by 'Kellogg's' will throw up quite different results depending on whether or not an apostrophe is inserted, and this can cause great confusion in the supply chain. Proposals to fix this problem can be surprisingly complicated [618]. There are also the issues of convergence discussed above; data might not be consistent across a system, even in theory. There are also the problems of timeliness, such as whether a product has been recalled.

   Now, many firms propose moving to RFID chips that contain a globally unique number: an item code rather than a product code. This may move name resolution upstream; rather than the shop's computer recognising that the customer has presented a packet of vitamin C at the checkout, it may go to the manufacturer to find this

out. Manufacturers push for this on safety grounds; they can then be sure that the product hasn't passed its sell-by date and has not been recalled. But this also increases their power over the supply chain; they can detect and stop gray-market trading that would otherwise undermine their ability to charge different prices in different towns.

9. *Don't get too smart. Phone numbers are much more robust than computer addresses*. Amen to that — but it's too often ignored by secure system designers. Bank account numbers are much easier to deal with than the public-key certificates which were once believed both necessary and sufficient to secure credit card payments online. I'll discuss specific problems of public key infrastructures in section 21.4.5.7.
10. *Some names are bound early, others not; and in general it is a bad thing to bind early if you can avoid it*. A prudent programmer will normally avoid coding absolute addresses or filenames as that would make it hard to upgrade or replace a machine. He will prefer to leave this to a configuration file or an external service such as DNS. (This is another reason not to put addresses in names.) It is true that secure systems often want stable and accountable names as any third-party service used for last minute resolution could be a point of attack. However, designers should read the story of Netgear, who got their home routers to find out the time using the Network Time Protocol from a server at the University of Wisconsin-Madison. Their product was successful; the university was swamped with hundreds of thousands of packets a second. Netgear ended up paying them $375,000 to maintain the time service for three years. Shortly afterwards, D-Link repeated the same mistake [304].

So Needham's ten principles for distributed naming apply fairly directly to distributed secure systems.

6.4.2 What Else Goes Wrong

Needham's principles, although very useful, are not sufficient. They were designed for a world in which naming systems could be designed and imposed at the system owner's convenience. When we move from distributed systems in the abstract to the reality of modern web-based (and interlinked) service industries, there is still more to say.

6.4.2.1 Naming and Identity

The most obvious difference is that the principals in security protocols may be known by many different kinds of name — a bank account number, a company registration number, a personal name plus a date of birth or a postal address,

a telephone number, a passport number, a health service patient number, or a userid on a computer system.

As I mentioned in the introductory chapter, a common mistake is to confuse naming with identity. *Identity* is when two different names (or instances of the same name) correspond to the same principal (this is known in the distributed systems literature as an *indirect name* or *symbolic link*). The classic example comes from the registration of title to real estate. It is very common that someone who wishes to sell a house uses a different name than they did at the time it was purchased: they might have changed their name on marriage, or after a criminal conviction. Changes in name usage are also common. For example, the DE Bell of the Bell-LaPadula system[2] wrote his name 'D. Elliot Bell' in 1973 on that paper; but he was always known as David, which is how he now writes his name too. A land-registration system must cope with a lot of identity issues like this.

The classic example of identity failure leading to compromise is check fraud. Suppose I steal a high-value check made out to Mr Elliott Bell. I then open an account in that name and cash it; banking law in both the USA and the UK absolves the bank of liability so long as it pays the check into an account of the same name. The modern procedure of asking people who open bank accounts for two proofs of address, such as utility bills, probably makes the bad guys' job easier; there are hundreds of utility companies, many of which provide electronic copies of bills that are easy to alter. The pre-9/11 system, of taking up personal references, may well have been better.

Moving to verifying government-issue photo-ID on account opening adds to the mix statements such as 'The Elliott Bell who owns bank account number 12345678 is the Elliott James Bell with passport number 98765432 and date of birth 3/4/56'. This may be seen as a symbolic link between two separate systems — the bank's and the passport office's. Note that the latter part of this 'identity' encapsulates a further statement, which might be something like 'The US passport office's file number 98765432 corresponds to the entry in birth register for 3/4/56 of one Elliott James Bell'. In general, names may involve several steps of recursion, and this gives attackers a choice of targets. For example, a lot of passport fraud is *pre-issue fraud*: the bad guys apply for passports in the names of genuine citizens who haven't applied for a passport already and for whom copies of birth certificates are easy enough to obtain. Postmortem applications are also common. Linden Labs, the operators of Second Life, introduced in late 2007 a scheme whereby you prove you're over 18 by providing the driver's license number or social security number of someone who is. Now a web search quickly pulls up such data for many people, such as the rapper Tupac Amaru Shakur; and yes, Linden Labs did accept Mr Shakur's license number — even through the license is expired, and

[2] I'll discuss this in Chapter 8, 'Multilevel Secure Systems'.

he's dead. Indeed, someone else managed to verify their age using Mohammed Atta's driver's license [389].

6.4.2.2 Cultural Assumptions

The assumptions that underlie names often change from one country to another. In the English-speaking world, people may generally use as many names as they please; a name is simply what you are known by. But some countries forbid the use of aliases, and others require them to be registered. This can lead to some interesting scams: in at least one case, a British citizen evaded pursuit by foreign tax authorities by changing his name. On a less exalted plane, women who pursue academic careers and change their name on marriage may wish to retain their former name for professional use, which means that the name on their scientific papers is different from their name on the payroll. This caused a row at my university which introduced a unified ID card system, keyed to payroll names, without support for aliases.

In general, many of the really intractable problems arise when an attempt is made to unify two local naming systems which turn out to have incompatible assumptions. As electronics invades everyday life more and more, and systems become linked up, conflicts can propagate and have unexpected remote effects. For example, one of the lady professors in the dispute over our university card was also a trustee of the British Library, which issues its own admission tickets on the basis of the name on the holder's home university library card.

Even human naming conventions are not uniform. Russians are known by a forename, a patronymic and a surname; Icelanders have no surname but are known instead by a given name followed by a patronymic if they are male and a matronymic if they are female. This causes problems when they travel. When US immigration comes across 'Maria Trosttadóttir' and learns that 'Trosttadóttir' isn't a surname or even a patronymic, their standard practice was to compel her to adopt as a surname a patronymic (say, 'Carlsson' if her father was called Carl). This causes unnecessary offence.

The biggest cultural divide is often thought to be that between the English speaking countries (where identity cards were long considered to be unacceptable on privacy grounds[3]) and the countries conquered by Napoleon or by the Soviets, where identity cards are the norm.

There are further subtleties. I know Germans who have refused to believe that a country could function at all without a proper system of population registration and ID cards, yet admit they are asked for their ID card only rarely (for example, to open a bank account or get married). Their card number can't be used as a name, because it is a document number and changes every time a new card is issued. A Swiss hotelier may be happy to register a German guest on

[3] unless they're called drivers' licences or health service cards!

sight of an ID card rather than a credit card, but if he discovers some damage after a German guest has checked out, he may be out of luck. And the British passport office will issue a citizen with more than one passport at the same time, if he says he needs them to make business trips to (say) Cuba and the USA; so our Swiss hotelier, finding that a British guest has just left without paying, can't rely on the passport number to have him stopped at the airport.

There are many other hidden assumptions about the relationship between governments and people's names, and they vary from one country to another in ways which can cause unexpected security failures.

6.4.2.3 Semantic Content of Names

Another hazard arises on changing from one type of name to another without adequate background research. A bank got sued after they moved from storing customer data by account number to storing it by name and address. They wanted to target junk mail more accurately, so they wrote a program to link up all the accounts operated by each of their customers. The effect for one poor customer was that the bank statement for the account he maintained for his mistress got sent to his wife, who divorced him.

Sometimes naming is simple, but sometimes it merely appears to be. For example, when I got a monthly ticket for the local swimming baths, the cashier simply took the top card off a pile, swiped it through a reader to tell the system it was now live, and gave it to me. I had been assigned a random name — the serial number on the card. Many US road toll systems work in much the same way. Sometimes a random, anonymous name can add commercial value. In Hong Kong, toll tokens for the Aberdeen tunnel could be bought for cash, or at a discount in the form of a refillable card. In the run-up to the transfer of power from Britain to Beijing, many people preferred to pay extra for the less traceable version as they were worried about surveillance by the new government.

Semantics of names can change. I once got a hardware store loyalty card with a random account number (and no credit checks). I was offered the chance to change this into a bank card after the store was taken over by the supermarket and the supermarket started a bank. (This appears to have ignored money laundering regulations that all new bank customers must be subjected to due diligence.)

Assigning bank account numbers to customers might have seemed unproblematic — but as the above examples show, systems may start to construct assumptions about relationships between names that are misleading and dangerous.

6.4.2.4 Uniqueness of Names

Human names evolved when we lived in small communities. We started off with just forenames, but by the late Middle Ages the growth of travel led

governments to bully people into adopting surnames. That process took a century or so, and was linked with the introduction of paper into Europe as a lower-cost and more tamper-resistant replacement for parchment; paper enabled the badges, seals and other bearer tokens, which people had previously used for road tolls and the like, to be replaced with letters that mentioned their names.

The mass movement of people, business and administration to the Internet in the decade after 1995 has been too fast to allow any such social adaptation. There are now many more people (and systems) online than we are used to dealing with. As I remarked at the beginning of this section, I used to be the only Ross Anderson I knew of, but thanks to search engines, I now know dozens of us. Some of us work in fields I've also worked in, such as software engineering and electric power distribution; the fact that I'm `www.ross-anderson.com` and `ross.anderson@iee.org` is down to luck — I got there first. (Even so, `rjanderson@iee.org` is somebody else.) So even the combination of a relatively rare name and a specialized profession is still ambiguous. Another way of putting this is that 'traditional usernames, although old-school-geeky, don't scale well to the modern Internet' [21].

Sometimes system designers are tempted to solve the uniqueness problem by just giving everything and everyone a number. This is very common in transaction processing, but it can lead to interesting failures if you don't put the uniqueness in the right place. A UK bank wanted to send £20m overseas, but the operator typed in £10m by mistake. To correct the error, a second payment of £10m was ordered. However, the sending bank's system took the transaction sequence number from the paperwork used to authorise it. Two payments were sent to SWIFT with the same date, payee, amount and sequence number — so the second was discarded as a duplicate [218].

6.4.2.5 Stability of Names and Addresses

Many names include some kind of address, yet addresses change. About a quarter of Cambridge phone book addresses change every year; with email, the turnover is probably higher. A project to develop a directory of people who use encrypted email, together with their keys, found that the main cause of changed entries was changes of email address [67]. (Some people had assumed it would be the loss or theft of keys; the contribution from this source was precisely zero.)

A serious problem could arise with IPv6. The security community assumes that v6 IP addresses will be stable, so that public key certificates can bind principals of various kinds to them. All sorts of mechanisms have been proposed to map real world names, addresses and even document content indelibly and eternally on to 128 bit strings (see, for example, [573]). The data communications community, on the other hand, assumes that IPv6 addresses

will change regularly. The more significant bits will change to accommodate more efficient routing algorithms, while the less significant bits will be used to manage local networks. These assumptions can't both be right.

Distributed systems pioneers considered it a bad thing to put addresses in names [912]. But in general, there can be multiple layers of abstraction with some of the address information at each layer forming part of the name at the layer above. Also, whether a namespace is better flat depends on the application. Often people end up with different names at the departmental and organizational level (such as `rja14@cam.ac.uk` and `ross.anderson@cl.cam.ac.uk` in my own case). So a clean demarcation between names and addresses is not always possible.

Authorizations have many (but not all) of the properties of addresses. Kent's Law tells designers that if a credential contains a list of what it may be used for, then the more things are on this list the shorter its period of usefulness. A similar problem besets systems where names are composite. For example, some online businesses recognize me by the combination of email address and credit card number. This is clearly bad practice. Quite apart from the fact that I have several email addresses, I have several credit cards. The one I use will depend on which of them is currently offering the best service or the biggest bribes.

There are many good reasons to use pseudonyms. It's certainly sensible for children and young people to use online names that aren't easily linkable to their real names. This is often advocated as a child-protection measure, although the number of children abducted and murdered by strangers in developed countries remains happily low and stable at about 1 per 10,000,000 population per year. A more serious reason is that when you go for your first job on leaving college aged 22, or for a CEO's job at 45, you don't want Google to turn up all your teenage rants. Many people also change email addresses from time to time to escape spam; I give a different email address to every website where I shop. Of course, there are police and other agencies that would prefer people not to use pseudonyms, and this takes us into the whole question of traceability online, which I'll discuss in Part II.

6.4.2.6 Adding Social Context to Naming

The rapid growth recently of social network sites such as Facebook points to a more human and scaleable way of managing naming. Facebook does not give me a visible username: I use my own name, and build my context by having links to a few dozen friends. (Although each profile does have a unique number, this does not appear in the page itself, just in URLs.) This fixes the uniqueness problem — Facebook can have as many Ross Andersons as care to turn up — and the stability problem (though at the cost of locking me into Facebook if I try to use it for everything).

Distributed systems folks had argued for some time that no naming system can be simultaneously globally unique, decentralized, and human-meaningful. It can only have two of those attributes (Zooko's triangle) [21]. In the past, engineers tanded to look for naming systems that were unique and meaningful, like URLs, or unique and decentralised, as with public-key certificates[4]. The innovation from sites like Facebook is to show on a really large scale that naming doesn't have to be unique at all. We can use social context to build systems that are both decentralised and meaningful — which is just what our brains evolved to cope with.

6.4.2.7 Restrictions on the Use of Names

The interaction between naming and society brings us to a further problem: some names may be used only in restricted circumstances. This may be laid down by law, as with the US *Social Security Number* (SSN) and its equivalents in many European countries. Sometimes it is a matter of marketing. I would rather not give out my residential address (or my home phone number) when shopping online, and I avoid businesses that demand them.

Restricted naming systems interact in unexpected ways. For example, it's fairly common for hospitals to use a patient number as an index to medical record databases, as this may allow researchers to use pseudonymous records for some limited purposes without much further processing. This causes problems when a merger of health maintenance organizations, or a new policy directive in a national health service, forces the hospital to introduce uniform names. In the UK, for example, the merger of two records databases — one of which used patient names while the other was pseudonymous — has raised the prospect of legal challenges to processing on privacy grounds.

Finally, when we come to law and policy, the definition of a name turns out to be unexpectedly tricky. Regulations that allow police to collect communications data — that is, a record of who called whom and when — are often very much more lax than the regulations governing phone tapping; in many countries, police can get this data just by asking the phone company. There was an acrimonious public debate in the UK about whether this enables them to harvest the URLs which people use to fetch web pages. URLs often have embedded in them data such as the parameters passed to search engines. Clearly there are policemen who would like a list of everyone who hit a URL such as http://www.google.com/search?q=cannabis+cultivation+UK; just as clearly, many people would consider such large-scale trawling to be an unacceptable invasion of privacy. The police argued that if they were limited to

[4]Carl Ellison, Butler Lampson and Ron Rivest went so far as to propose the SPKI/SDSI certificate system in which naming would be relative, rather than fixed with respect to central authority. The PGP web of trust worked informally in the same way.

monitoring IP addresses, they could have difficulties tracing criminals who use transient IP addresses. In the end, Parliament resolved the debate when it passed the Regulation of Investigatory Powers Act in 2000: the police just get the identity of the machine under the laxer regime for communications data.

6.4.3 Types of Name

The complexity of naming appears at all levels — organisational, technical and political. I noted in the introduction that names can refer not just to persons (and machines acting on their behalf), but also to organizations, roles ('the officer of the watch'), groups, and compound constructions: *principal in role* — Alice as manager; *delegation* — Alice for Bob; *conjunction* — Alice and Bob. Conjunction often expresses implicit access rules: 'Alice acting as branch manager plus Bob as a member of the group of branch accountants'.

That's only the beginning. Names also apply to services (such as NFS, or a public key infrastructure) and channels (which might mean wires, ports, or crypto keys). The same name might refer to different roles: 'Alice as a computer game player' ought to have less privilege than 'Alice the system administrator'. The usual abstraction used in the security literature is to treat them as different principals. This all means that there's no easy mapping between names and principals.

Finally, there are functional tensions which come from the underlying business processes rather than from system design. Businesses mainly want to get paid, while governments want to identify people uniquely. In effect, business wants a credit card number while government wants a passport number. Building systems which try to be both — as many governments are trying to encourage — is a tar-pit. There are many semantic differences. You can show your passport to a million people, if you wish, but you had better not try that with a credit card. Banks want to open accounts for anyone who turns up with some money; governments want them to verify people's identity carefully in order to discourage money laundering. The list is a long one.

6.5 Summary

Many secure distributed systems have incurred huge costs, or developed serious vulnerabilities, because their designers ignored the basic lessons of how to build (and how not to build) distributed systems. Most of these lessons are still valid, and there are more to add.

A large number of security breaches are concurrency failures of one kind or another; systems use old data, make updates inconsistently or in the wrong order, or assume that data are consistent when they aren't and can't be. Knowing the right time is harder than it seems.

Fault tolerance and failure recovery are critical. Providing the ability to recover from security failures, and random physical disasters, is the main purpose of the protection budget for many organisations. At a more technical level, there are significant interactions between protection and resilience mechanisms. Byzantine failure — where defective processes conspire, rather than failing randomly — is an issue, and interacts with our choice of cryptographic tools. There are many different flavors of redundancy, and we have to use the right combination. We need to protect not just against failures and attempted manipulation, but also against deliberate attempts to deny service which may often be part of larger attack plans.

Many problems also arise from trying to make a name do too much, or making assumptions about it which don't hold outside of one particular system, or culture, or jurisdiction. For example, it should be possible to revoke a user's access to a system by cancelling their user name without getting sued on account of other functions being revoked. The simplest solution is often to assign each principal a unique identifier used for no other purpose, such as a bank account number or a system logon name. But many problems arise when merging two systems that use naming schemes that are incompatible for some reason. Sometimes this merging can even happen by accident — an example being when two systems use a common combination such as 'name plus date of birth' to track individuals, but in different ways.

Research Problems

In the research community, secure distributed systems tend to have been discussed as a side issue by experts on communications protocols and operating systems, rather than as a discipline in its own right. So it is a relatively open field, and one still holds much promise.

There are many technical issues which I've touched on in this chapter, such as how we design secure time protocols and the complexities of naming. But perhaps the most important research problem is to work out how to design systems that are resilient in the face of malice, that degrade gracefully, and whose security can be recovered simply once the attack is past. This may mean revisiting the definition of convergent applications. Under what conditions can one recover neatly from corrupt security state?

What lessons do we need to learn from the onset of phishing and keylogging attacks on electronic banking, which mean that at any given time a small (but nonzero) proportion of customer accounts will be under criminal control? Do we have to rework recovery (which in its classic form explores how to rebuild databases from backup tapes) into resilience, and if so how do we handle the tensions with the classic notions of atomicity, consistency,