CHAPTER 10

Banking and Bookkeeping

The arguments of lawyers and engineers pass through one another like angry ghosts.

— Nick Bohm, Brian Gladman and Ian Brown [201]

Computers are not (yet?) capable of being reasonable any more than is a Second Lieutenant.

— Casey Schaufler

Against stupidity, the Gods themselves contend in vain.

— JC Friedrich von Schiller

10.1 Introduction

Banking systems range from cash machine networks and credit card processing, both online and offline, through high-value interbank money transfer systems, to the back-end bookkeeping systems that keep track of it all and settle up afterwards. There are specialised systems for everything from stock trading to bills of lading; and large companies have internal bookkeeping and cash management systems that duplicate many of the functions of a bank.

Such systems are important for a number of reasons. First, an understanding of transaction processing is a prerequisite for tackling the broader problems of electronic commerce and fraud. Many dotcom firms fell down badly on elementary bookkeeping; in the rush to raise money and build web sites, traditional business discipline was ignored. The collapse of Enron led to stiffened board-level accountability for internal control; laws such as

Sarbanes-Oxley and Gramm-Leach-Bliley now drive much of the investment in information security. When you propose protection mechanisms to a client, one of the first things you're likely to be asked is the extent to which they'll help directors of the company discharge their fiduciary responsibilities to shareholders.

Second, bookkeeping was for many years the mainstay of the computer industry, with banking its most intensive application area. Personal applications such as web browsers and Office might now run on more machines, but accounting is still the critical application for the average business. So the protection of bookkeeping systems is of great practical importance. It also gives us a well-understood model of protection in which confidentiality plays little role, but where the integrity of records (and their immutability once made) is of paramount importance.

Third, transaction processing systems — whether for small debits such as $50 cash machine withdrawals, or multimillion dollar wire transfers — were the application that launched commercial cryptology. Banking applications drove the development not just of encryption algorithms and protocols, but also of the supporting technology such as tamper-resistant cryptographic processors. These processors provide an important and interesting example of a trusted computing base that is quite different from the hardened operating systems discussed in the context of multilevel security. Many instructive mistakes were first made (or at least publicly documented) in the area of commercial cryptography. The problem of how to interface crypto with access control was studied by financial cryptographers before any others in the open research community.

Finally, banking systems provide another example of multilateral security — but aimed at authenticity rather than confidentiality. A banking system should prevent customers from cheating each other, or the bank; it should prevent bank employees from cheating the bank, or its customers; and the evidence it provides should be sufficiently strong that none of these principals can get away with falsely accusing another principal of cheating.

In this chapter, I'll first describe the bookkeeping systems used to keep track of assets despite occasional corrupt staff; these are fairly typical of accounting systems used by other companies too. I'll then describe the banks' principal international funds-transfer systems; similar systems are used to settle securities transactions and to manage trade documents such as bills of lading. Next, I'll describe ATM systems, which are increasingly the public face of banking, and whose technology has been adopted in applications such as utility meters; and then I'll tell the story of credit cards, which have become the main payment mechanism online. I'll then move on to more recent technical advances, including the smartcards recently introduced in Europe, RFID credit

cards, and nonbank payment services such as PayPal. I'll wrap up with some points on money laundering, and what controls really work against fraud.

10.1.1 The Origins of Bookkeeping

Bookkeeping appears to have started in the Neolithic Middle East in about 8500 BC, just after the invention of agriculture [1122]. When people started to produce surplus food, they started to store and trade it. Suddenly they needed a way to keep track of which villager had put how much in the communal warehouse. To start with, each unit of food (sheep, wheat, oil, . . .) was represented by a clay token, or *bulla*, which was placed inside a clay envelope and sealed by rolling it with the pattern of the warehouse keeper. (See Figure 10.1.) When the farmer wanted to get his food back, the seal was broken by the keeper in the presence of a witness. (This may be the oldest known security protocol.) By about 3000BC, this had led to the invention of writing [1018]; after another thousand years, we find equivalents of promissory notes, bills of lading, and so on. At about the same time, metal ingots started to be used as an intermediate commodity, often sealed inside a bulla by an assayer. In 700BC, Lydia's King Croesus started stamping the metal directly and thus invented coins [1045]; by the Athens of Pericles, there were a number of wealthy individuals in business as bankers [531].

Figure 10.1: Clay envelope and its content of tokens representing 7 jars of oil, from Uruk, present day Iraq, ca. 3300 BC (Courtesy Denise Schmandt-Besserat and the Louvre Museum)

The next significant innovation dates to late medieval times. As the dark ages came to a close and trade started to grow, some businesses became too large for a single family to manage. The earliest of the recognisably modern banks date to this period; by having branches in a number of cities, they could finance trade efficiently. But as the economy grew, it was necessary to hire managers from outside, and the owner's family could not supervise them closely. This brought with it an increased risk of fraud, and the mechanism that evolved to control it was *double-entry bookkeeping*. People used to think this was invented in Italy sometime in the 1300s, though the first book on it did not appear until 1494, after the printing press came along [355]. Recently, however, historians have found double-entry records created by Jewish merchants in twelfth-century Cairo, and it's now believed that the Italians learned the technique from them [1140].

10.1.2 Double-Entry Bookkeeping

The idea behind double-entry bookkeeping is extremely simple, as with most hugely influential ideas. Each transaction is posted to two separate books, as a credit in one and a debit in the other. For example, when a firm sells a customer $100 worth of goods on credit, it posts a $100 credit on the Sales account, and a $100 debit onto the Receivables account. When the customer pays the money, it will credit the Receivables account (thereby reducing the asset of money receivable), and debit the Cash account. (The principle taught in accountancy school is 'debit the receiver, credit the giver'.) At the end of the day, the books should *balance*, that is, add up to zero; the assets and the liabilities should be equal. (If the firm has made some profit, then this is a liability to the shareholders.) 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).

By suitable design of the ledger system, we can see to it that each shop, or branch, can be balanced separately. Thus each cashier will balance her cash tray before locking it in the vault overnight; the debits in the cash legder should exactly balance the physical banknotes she's collected. So most frauds need the collusion of two or more members of staff; and this principle of *split responsibility*, also known as *dual control*, is complemented by audit. Not only are the books audited at year end, but there are random audits too; a team of inspectors may descend on a branch at no notice and insist that all the books are balanced before the staff go home.

10.1.3 A Telegraphic History of E-commerce

Many of the problems afflicting e-businesses stem from the popular notion that e-commerce is something completely new, invented in the mid-1990s. This is simply untrue.

Various kinds of visual signalling were deployed from classical times, including heliographs (which used mirrors to flash sunlight at the receiver), semaphones (which used the positions of moving arms to signal letters and numbers) and flags. Land-based systems sent messages along chains of beacon towers, and naval systems relayed them between ships. To begin with, their use was military, but after the Napoleonic War the French government opened its heliograph network to commercial use. Very soon the first frauds were carried out. For two years up till they were discovered in 1836, two bankers bribed an operator to signal the movements of the stock market to them covertly by making errors in transmissions that they could observe from a safe distance. Other techniques were devised to signal the results of horseraces. Various laws were passed to criminalise this kind of activity but they were ineffective. The only solution for the bookies was to 'call time' by a clock, rather than waiting for the result and hoping that they were the first to hear it.

From the 1760's to the 1840's, the electric telegraph was developed by a number of pioneers, of whom the most influential was Samuel Morse. He persuaded Congress in 1842 to fund an experimental line from Washington to Baltimore; this so impressed people that serious commercial investment started, and by the end of that decade there were 12,000 miles of line operated by 20 companies. This was remarkably like the Internet boom of the late 1990's.

Banks were the first big users of the telegraph, and they decided that they needed technical protection mechanisms to prevent transactions being altered by crooked operators en route. (I discussed the *test key* systems they developed for the purpose in section 5.2.4.) Telegrams were also used to create national markets. For the first time, commodity traders in New York could find out within minutes what prices had been set in auctions in Chicago, and fishing skippers arriving in Boston could find out the price of cod in Gloucester. The history of the period shows that most of the concepts and problems of e-commerce were familiar to the Victorians [1215]. How do you know who you're speaking to? How do you know if they're trustworthy? How do you know whether the goods will be delivered, and whether payments will arrive? The answers found in the nineteenth century involved intermediaries — principally banks who helped business manage risk using instruments such as references, guarantees and letters of credit.

10.2 How Bank Computer Systems Work

Banks were among the first large organizations to use computers for book-keeping. They started in the late 1950s and early 1960s with applications such as check processing, and once they found that even the slow and expensive computers of that era were much cheaper than armies of clerks, they proceeded to automate most of the rest of their back-office operations during the

1960s and 1970s. The 1960s saw banks offering automated payroll services to their corporate customers, and by the 1970s they were supporting business-to-business e-commerce based on *electronic data interchange* (EDI), whereby firms from General Motors to Marks and Spencer built systems that enabled them to link up their computers to their suppliers' so that goods could be ordered automatically. Travel agents built similar systems to order tickets in real time from airlines. ATMs arrived en masse in the 1970s, and online banking systems in the 1980s; web-based banking followed in the 1990s. Yet the fancy front-end systems still rely on traditional back-office automation for maintaining account data and performing settlement.

Computer systems used for bookkeeping typically claim to implement variations on the double-entry theme. But the quality of control is highly variable. The double-entry features may be just a skin in the user interface, while the underlying file formats have no integrity controls. And even if the ledgers are all kept on the same system, someone with root access — or with physical access and a debugging tool — may be able to change the 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. Despite all these problems, the law in most developed countries requires companies to have effective internal controls, and makes the managers responsible for them. Such laws are the main drivers of investment in information security mechanisms, but they also a reason for much wasted investment. So we need to look at the mechanics of electronic bookkeeping in a more detail.

A typical banking system has a number of data structures. There is an *account master file* which contains each customer's current balance together with previous transactions for a period of perhaps ninety days; a number of *ledgers* which track cash and other assets on their way through the system; various *journals* which hold transactions that have been received from teller stations, cash machines, check sorters and so on, but not yet entered in the ledgers; and an *audit trail* that records which staff member did what and when.

The processing software that acts on these data structures will include a suite of overnight batch processing programs, which apply the transactions from the journals to the various ledgers and the account master file. The online processing will include a number of modules which post transactions to the relevant combinations of ledgers. So when a customer pays $100 into his savings account the teller will make a transaction which records a credit to the customer's savings account ledger of $100 while debiting the same amount to the cash ledger recording the amount of money in the drawer. The fact that all the ledgers should always add up to zero provides an important check; if the bank (or one of its branches) is ever out of balance, an alarm will go off and people will start looking for the cause.

The invariant provided by the ledger system is checked daily during the overnight batch run, and means that a programmer who wants to add to his own account balance will have to take the money from some other account, rather than just creating it out of thin air by tweaking the account master file. Just as in a traditional business one has different ledgers managed by different clerks, so in a banking data processing shop there are different programmers in charge of different subsystems. In addition, all code is subjected to scrutiny by an internal auditor, and to testing by a separate test department. Once it has been approved, it will be run on a production machine that does not have a development environment, but only approved object code and data.

10.2.1 The Clark-Wilson Security Policy Model

Although such systems had been in the field since the 1960s, a formal model of their security policy was only introduced in 1987, by Dave Clark and Dave Wilson (the former a computer scientist, and the latter an accountant) [295]. In their model, some data items are constrained so that they can only be acted on by a certain set of transformation procedures.

More formally, there are special procedures whereby data can be input — turned from an *unconstrained data item*, or UDI, into a *constrained data item*, or CDI; *integrity verification procedures* (IVP's) to check the validity of any CDI (e.g., that the books balance); and *transformation procedures* (TPs) which may be thought of in the banking case as transactions which preserve balance. In the general formulation, they maintain the integrity of CDIs; they also write enough information to an append-only CDI (the audit trail) for transactions to be reconstructed. Access control is by means of triples *(subject, TP, CDI)*, which are so structured that a dual control policy is enforced. In the formulation in [27]:

1. the system will have an IVP for validating the integrity of any CDI;
2. the application of a TP to any CDI must maintain its integrity;
3. a CDI can only be changed by a TP;
4. subjects can only initiate certain TPs on certain CDIs;
5. triples must enforce an appropriate separation-of-duty policy on subjects;
6. certain special TPs on UDIs can produce CDIs as output;
7. each application of a TP must cause enough information to reconstruct it to be written to a special append-only CDI;
8. the system must authenticate subjects attempting to initiate a TP;
9. the system must let only special subjects (i.e., security officers) make changes to authorization-related lists.

A number of things bear saying about Clark-Wilson.

First, unlike Bell-LaPadula, Clark-Wilson involves maintaining state. Even disregarding the audit trail, this is usually necessary for dual control as you have to keep track of which transactions have been partially approved — such as those approved by only one manager when two are needed. If dual control is implemented using access control mechanisms, it typically means holding partially approved transactions in a special journal file. This means that some of the user state is actually security state, which in turn makes the trusted computing base harder to define. If it is implemented using crypto instead, such as by having managers attach digital signatures to transactions of which they approve, then there can be problems managing all the partially approved transactions so that they get to a second approver in time.

Second, the model doesn't do everything. It captures the idea that state transitions should preserve an invariant such as balance, but not that state transitions should be correct. Incorrect transitions, such as paying into the wrong bank account, are not prevented by this model.

Third, Clark-Wilson ducks the hardest question, namely: how do we control the risks from dishonest staff? Rule 5 says that 'an appropriate separation of duty policy' must be supported, but nothing about what this means. Indeed, it's very hard to find any systematic description in the accounting literature of how you design internal controls — it's something that auditors tend to learn on the job. Companies' internal controls tend to evolve over time in response to real or feared incidents, whether in the company's own experience or its auditors'. In the next section, I try to distill into a few principles the experience gained from several years working at the coalface in banking and consultancy, and more recently on our university's finance and other committees.

10.2.2 Designing Internal Controls

Over the years, a number of standards have been put forward by the accountancy profession, by stock markets and by banking regulators, about how bookkeeping and internal control systems should be designed. In the USA, for example, there is the *Committee of Sponsoring Organizations* (COSO), a group of U.S. accounting and auditing bodies [318]. This self-regulation failed to stop the excesses of the dotcom era, and following the collapse of Enron there was intervention from U.S. lawmakers in the form of the Sarbanes-Oxley Act of 2002. It protects whistleblowers (the main source of information on serious insider fraud), and its section 404 makes managers responsible for maintaining 'adequate internal control structure and procedures for financial reporting'. It also demands that auditors attest to the management's assessment of these controls and disclose any 'material weaknesses'. CEOs also have to certify the truthfulness of financial statements. There was also the Gramm-Leach-Bliley Act of 1999, which liberalised bank regulation in many respects but

which obliged banks to have security mechanisms to protect information from foreseeable threats in security and integrity. Along with HIPAA in the medical sector, Gramm-Leach-Bliley and Sarbanes-Oxley have driven much of the investment in information security and internal control over the early years of the 21st century. (Other countries have equivalents; in the UK it's the Turnbull Guidance from the Financial Reporting Council.) I'll return to them and look in more detail at the policy aspects in Part III.

In this section, my concern is with the technical aspects. Modern risk management systems typically require a company to identify and assess its risks, and then build controls to mitigate them. A company's risk register might contain many pages of items such as 'insider makes large unauthorised bank transaction'. Some of these will be mitigated using non-technical measures such as insurance, but others will end up in your lap. So how do you engineer away a problem like this?

There are basically two kinds of separation of duty policy: dual control and functional separation.

In dual control, two or more staff members must act together to authorize a transaction. The classic military example is in nuclear command systems, which may require two officers to turn their keys simultaneously in consoles that are too far apart for any single person to reach both locks. I'll discuss nuclear command and control further in a later chapter. The classic civilian example is when a bank issues a letter of guarantee, which will typically undertake to carry the losses should a loan made by another bank go sour. Guarantees are particularly prone to fraud; if you can get bank A to guarantee a loan to your business from bank B, then bank B is supervising your account while bank A's money is at risk. A dishonest businessmen with a forged or corruptly-obtained guarantee can take his time to plunder the loan account at bank B, with the alarm only being raised when he absconds and bank B asks bank A for the money. If a single manager could issue such an instrument, the temptation would be strong. I'll discuss this further in section 10.3.2.

With functional separation of duties, two or more different staff members act on a transaction at different points in its path. The classic example is corporate purchasing. A manager takes a purchase decision and tells the purchasing department; a clerk there raises a purchase order; the store clerk records the goods' arrival; an invoice arrives at accounts; the accounts clerk correlates it with the purchase order and the stores receipt and raises a check; and the accounts manager signs the check.

However, it doesn't stop there. The manager now gets a debit on her monthly statement for that internal account, her boss reviews the accounts to make sure the division's profit targets are likely to be met, the internal audit department can descend at any time to audit the division's books, and when the external auditors come in once a year they will check the books of a randomly selected

sample of departments. Finally, when frauds are discovered, the company's lawyers may make vigorous efforts to get the money back.

So the model can be described as *prevent — detect — recover*. The level of reliance placed on each of these three legs will depend on the application. Where detection may be delayed for months or years, and recovery may therefore be very difficult — as with corrupt bank guarantees — it is prudent to put extra effort into prevention, using techniques such as dual control. Where prevention is hard, you should see to it that detection is fast enough, and recovery vigorous enough, to provide a deterrent effect. The classic example here is that bank tellers can quite easily take cash, so you need to count the money every day and catch them afterwards.

Bookkeeping and management control are not only one of the earliest security systems; they also have given rise to much of management science and civil law. They are entwined with a company's business processes, and exist in its cultural context. In Swiss banks, there are two managers' signatures on almost everything, while Americans are much more relaxed. In most countries' banks, staff get background checks, can be moved randomly from one task to another, and are forced to take holidays at least once a year. This would not be acceptable in the typical university — but in academia the opportunities for fraud are much less.

Designing an internal control system is hard because it's a highly interdisciplinary problem. The financial controllers, the personnel department, the lawyers, the auditors and the systems people all come at the problem from different directions, offer partial solutions, fail to understand each other's control objectives, and things fall down the hole in the middle. Human factors are very often neglected, and systems end up being vulnerable to helpful subordinates or authoritarian managers who can cause dual control to fail. It's important not just to match the controls to the culture, but also motivate people to use them; for example, in the better run banks, management controls are marketed to staff as a means of protecting them against blackmail and kidnapping.

Security researchers have so far focused on the small part of the problem which pertains to creating dual control (or in general, where there are more than two principals, *shared control*) systems. Even this is not at all easy. For example, rule 9 in Clark-Wilson says that security officers can change access rights — so what's to stop a security officer creating logons for two managers and using them to send all the bank's money to Switzerland?

In theory you could use cryptography, and split the signing key between two or more principals. In a Windows network, the obvious way to manage things is to put users in separately administered domains. With a traditional banking system using the mainframe operating system MVS, you can separate duties between the sysadmin and the auditor; the former can do anything he wishes, except finding out which of his activities the latter is monitoring [159]. But in real life, dual control is hard to do end-to-end because there are

many system interfaces that provide single points of failure, and in any case split-responsibility systems administration is tedious.

So the practical answer is that most bank sysadmins are in a position to do just this type of fraud. Some have tried, and where they fall down is when the back-office balancing controls set off the alarm after a day or two and money laundering controls stop him getting away with very much. I'll discuss this further in section 10.3.2. The point to bear in mind here is that serial controls along the *prevent — detect — recover* model are usually more important than shared control. They depend ultimately on some persistent state in the system and are in tension with programmers' desire to keep things simple by making transactions atomic.

There are also tranquility issues. For example, could an accountant knowing that he was due to be promoted to manager tomorrow end up doing both authorizations on a large transfer? A technical fix for this might involve a Chinese Wall mechanism supporting a primitive '*X* may do *Y* only if he hasn't done *Z*' ('A manager can confirm a payment only if his name doesn't appear on it as the creator'). So we would end up with a number of exclusion and other rules involving individuals, groups and object labels; once the number of rules becomes large (as it will in a real bank) we would need a systematic way of examining this rule set and verifying that it doesn't have any horrible loopholes.

In the medium term, banking security policy — like medical security policy — may find its most convenient expression in using role based access control, although this will typically be implemented in banking middleware rather than in an underlying platform such as Windows or Linux. Real systems will need to manage separation-of-duty policies with both parallel elements, such as dual control, and serial elements such as functional separation along a transaction's path. This argues for the access control mechanisms being near to the application. But then, of course, they are likely to be more complex, proprietary, and not so well studied as the mechanisms that come with the operating system.

One really important aspect of internal control in banking — and in systems generally — is to minimise the number of 'sysadmins', that is, of people with complete access to the whole system and the ability to change it. For decades now, the standard approach has been to keep development staff quite separate from live production systems. A traditional bank in the old days would have two mainframes, one to run the live systems, with the other being a backup machine that was normally used for development and testing. Programmers would create new software that would be tested by a test department and subject to source code review by internal auditors; once approved this would be handed off to a change management department that would install it in the live system at the next upgrade. The live system would be run by an operations

team with no access to compilers, debuggers or other tools that would let them alter live code or data.

In theory this prevents abuse by programmers, and in practice it can work fairly well. However there are leaks. First, there are always some sysadmins who need full access in order to do their jobs; and second, there are always emergencies. The ATM system goes down at the weekend, and the ATM team's duty programmer is given access to the live system from home so she can fix the bug. You audit such accesses as well as you can, but it's still inevitable that your top sysadmins will be so much more knowledgeable than your auditors that they could do bad things if they really wanted to. Indeed, at banks I've helped with security, you might find that there are thirty or forty people whom you just have to trust — the CEO, the chief dealer, the top sysadmins and a number of others. It's important to know who these people are, and to minimise their numbers. Pay them well — and watch discreetly to see if they start spending even more.

A final remark on dual control is that it's often not adequate for transactions involving more than one organization, because of the difficulties of dispute resolution: 'My two managers say the money was sent!' 'But my two say it wasn't!'

10.2.3 What Goes Wrong

Theft can take a variety of forms, from the purely opportunist to clever insider frauds; but the experience is that most thefts from the average company are due to insiders. There are many surveys; a typical one, by accountants Ernst and Young, reports that 82% of the worst frauds were committed by employees; nearly half of the perpetrators had been there over five years and a third of them were managers [1162].

Typical computer crime cases include:

- Paul Stubbs, a password reset clark at HSBC, conspired with persons unknown to change the password used by AT&T to access their bank account with HSBC. The new password was used to transfer £11.8 million — over $20 million — to offshore companies, from which it was not recovered. Stubbs was a vulnerable young man who had been employed as a password reset clerk after failing internal exams; the court took mercy on him and he got away with five years [975]. It was alleged that an AT&T employee had conspired to cover up the transactions, but that gentleman was acquitted.
- 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). This was a workaround added to the dual control system to deal with

transactions that got lost or otherwise couldn't be balanced immediately. As it was a potential vulnerability, the bank had a rule that suspense accounts would be investigated if they were not cleared within three days. One of the clerks exploited this by setting 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 dollars. (The bank negligently ignored a regulatory requirement that all staff take at least ten consecutive days' vacation no more than fifteen months from the last such vacation.) In the end, 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 school, complete with staff whose salaries were paid into her own bank account. It was only discovered by accident when someone noticed that different records gave the authority different numbers of schools.
- A bank clerk in Hastings, England, 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 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 started stuffing the cash he'd stolen in brown envelopes through the branch's letter box at night. As people don't normally give money to banks, the branch manager finally realized that something was seriously wrong.

Volume crime — such as card fraud — often depends on liability rules. Where banks can tell customers who complain of fraud to get lost (as in much of Europe), bank staff know that complaints won't be investigated properly or at all, and get careless. Things are better in the USA where Regulation E places the onus of proof in disputed transaction cases squarely on the bank. I'll discuss this in detail in section 10.4 below.

All the really large frauds — the cases over a billion dollars — have involved lax internal controls. The collapse of Barings Bank is a good example: managers failed to control rogue trader Nick Leeson, blinded by greed for the bonuses his apparent trading profits earned them. The same holds true for other financial sector frauds, such as the Equity Funding scandal, in which an insurance company's management created thousands of fake people on their computer system, insured them, and sold the policies on to reinsurers; and frauds in

other sectors such as Robert Maxwell's looting of the Daily Mirror newspaper pension funds in Britain. (For a collection of computer crime case histories, see Parker [1005].) Either the victim's top management were grossly negligent, as in the case of Barings, or perpetrated the scam, as with Equity Funding and Maxwell.

The auditors are also a problem. On the one hand, they are appointed by the company's managers and are thus extremely bad at detecting frauds in which the managers are involved; so the assurance that shareholders get is less than many might have thought. (The legal infighting following the collapse of Enron destroyed its auditors Arthur Andersen and thus reduced the 'big five' audit firms to the 'big four'; now auditors go out of their way to avoid liability for fraud.) Second, there were for many years huge conflicts of interest, as accountants offered cheap audits in order to get their foot in the door, whereupon they made their real money from systems consultancy. (This has been greatly restricted since Enron.) Third, the big audit firms have their own list of favourite controls, which often bear little relationship to the client's real risks, and may even make matters worse. For example, our university's auditors nag us every year to get all our staff to change their passwords every month. This advice is wrong, for reasons explained in Chapter 2 — so every year we point this out and challenge them to justify their advice. But they seem incapable of learning, and they have no incentive to: they can be expected to nitpick, and to ignore any evidence that a particular nit is unhelpful until long after the evidence has become overwhelming. While failing to disclose a material weakness could get them into trouble, at least in the USA, the nitpicking has turned into a bonanza for them. It's reckoned that the auditors' gold-plating of the Sarbanes-Oxley requirements is costing the average U.S. listed company $2.4m a year in audit fees, plus 70,000 hours of internal work to ensure compliance; the total cost of SOX could be as much as $1.4 trillion [412]. (My own advice, for what it's worth, is to never use a big-four accountant; smaller firms are cheaper, and a study done by my student Tyler Moore failed to find any evidence that companies audited by the Big Four performed better on the stock market.)

Changing technology also has a habit of eroding controls, which therefore need constant attention and maintenance. For example, thanks to new systems for high-speed processing of bank checks, banks in California stopped a few years ago from honoring requests by depositors that checks have two signatures. Even when a check has imprinted on it 'Two Signatures Required', banks will honor that check with only one signature [1086]. This might seem to be a problem for the customer's security rather than the bank's, but bank checks can also be at risk and if something goes wrong even with a merchant transaction then the bank might still get sued.

The lessons to be learned include:

- it's not always obvious which transactions are security sensitive;
- maintaining a working security system can be hard in the face of a changing environment;
- if you rely on customer complaints to alert you to fraud, you had better listen to them;
- there will always be people in positions of relative trust who can get away with a scam for a while;
- no security policy will never be completely rigid. There will always have to be workarounds for people to cope with real life;
- these workarounds naturally create vulnerabilities. So the lower the transaction error rate, the better.

There will always be residual risks. Managing these residual risks remains one of the hardest and most neglected of jobs. It means not just technical measures, such as involving knowledgeable industry experts, auditors and insurance people in the detailed design, and iterating the design once some loss history is available. It also means training managers, auditors and others to detect problems and react to them appropriately. I'll revisit this in Part III.

The general experience of banks in the English-speaking world is that some 1% of staff are sacked each year. The typical offence is minor embezzlement with a loss of a few thousand dollars. No-one has found an effective way of predicting which staff will go bad; previously loyal staff can be thrown off the rails by shocks such as divorce, or may over time develop a gambling or alcohol habit. Losing a few hundred tellers a year is simply seen as a cost of doing business. What banks find very much harder to cope with are incidents in which senior people go wrong — indeed, in several cases within my experience, banks have gone to great lengths to avoid admitting that a senior insider was bent. And risks that managers are unwilling to confront, they are often unable to control. No-one at Barings even wanted to think that their star dealer Leeson might be a crook; and pop went the bank.

Finally, it's not enough, when doing an audit or a security investigation, to merely check that the books are internally consistent. It's also important to check that the correspond to external reality. This was brought home to the accounting profession in 1938 with the collapse of McKesson and Robbins, a large, well-known drug and chemical company with reported assets of $100m[1]. It turned out that 20% of the recorded assets and inventory were nonexistent. The president, Philip Musica, turned out to be an impostor with

[1] In 2007 dollars, that's $1.4bn if you deflate by prices, $3bn if you deflate by unskilled wages and over $15bn by share of GDP.

a previous fraud conviction; with his three brothers, he inflated the firm's figures using a fake foreign drug business involving a bogus shipping agent and a fake Montreal bank. The auditors, who had accepted the McKesson account without making enquiries about the company's bosses, had failed to check inventories, verify accounts receivable with customers, or think about separation of duties within the company [1082]. The lessons from that incident clearly weren't learned well enough, as the same general things continue to happen regularly and on all sorts of scales from small firms and small branches of big ones, to the likes of Enron.

So if you ever have responsibility for security in a financial (or other) firm, you should think hard about which of your managers could defraud your company by colluding with customers or suppliers. Could a branch manager be lending money to a dodgy business run by his cousin against forged collateral? Could he have sold life-insurance policies to nonexistent people and forged their death certificates? Could an operations manager be taking bribes from a supplier? Could one of your call-center staff be selling customer passwords to the Mafia? Lots of things can and do go wrong; you have to figure out which of them matter, and how you get to find out. Remember: a trusted person is one who can damage you. Who can damage you, and how? This is the basic question that a designer of internal controls must be constantly asking.

10.3 Wholesale Payment Systems

When people think of electronic bank fraud, they often envisage a Hollywood scene in which crafty Russian hackers break a bank's codes and send multi-million dollar wire transfers to tax havens. Systems for transferring money electronically are indeed an occasional target of sophisticated crime, and have been for a century and a half, as I noted earlier in section 5.2.4 when I discussed test key systems.

By the early 1970s, bankers started to realise that this worthy old Victorian system was due for an overhaul.

First, most test-key systems were vulnerable in theory at least to cryptanalysis; someone who observed a number of transactions could gradually work out the key material.

Second, although the test key tables were kept in the safe, there was nothing really to stop staff members working out tests for unauthorised messages at the same time as a test for an authorised message. In theory, you might require that two staff members retrieve the tables from the safe, sit down at a table facing each other and perform the calculation. However, in practice people would work sequentially in a corner (the tables were secret, after all) and even if you could compel them to work together, a bent employee might mentally compute the test on an unauthorized message while overtly computing the

test on an authorized one. So, in reality, test key schemes didn't support dual control. Having tests computed by one staff member and checked by another doubled the risk rather than reducing it. (There are ways to do dual control with manual authenticators, such as by getting the two staff members to work out different codes using different tables; and there are other techniques, used in the control of nuclear weapons, which I'll discuss in 13.4.)

Third, there was a big concern with cost and efficiency. There seemed little point in having the bank's computer print out a transaction in the telex room, having a test computed manually, then composing a telex to the other bank, then checking the test, and then entering it into the other bank's computer. Errors were much more of a problem than frauds, as the telex operators introduced typing errors. Customers who got large payments into their accounts in error sometimes just spent the money, and in one case an erroneous recipient spent some of his windfall on clever lawyers, who helped him keep it. This shocked the industry. Surely the payments could flow directly from one bank's computer to another?

10.3.1 SWIFT

The Society for Worldwide International Financial Telecommunications (SWIFT) was set up in the 1970s by a consortium of banks to provide a more secure and efficient means than telex of sending payment instructions between member banks. It can be thought of as an email system with built-in encryption, authentication and non-repudiation services. It's important not just because it's used to ship trillions of dollars round the world daily, but because its design has been copied in systems processing many other kinds of intangible asset, from equities to bills of lading.

The design constraints are interesting. The banks did not wish to trust SWIFT, in the sense of enabling dishonest employees there to forge transactions. The authenticity mechanisms had to be independent of the confidentiality mechanisms, since at the time a number of countries (such as France) forbade the civilian use of cryptography for confidentiality. The non-repudiation functions had to be provided without the use of digital signatures, as these hadn't been invented yet. Finally, the banks had to be able to enforce Clark-Wilson type controls over interbank transactions. (Clark-Wilson also hadn't been invented yet but its components, dual control, balancing, audit and so on, were well enough established.)

The SWIFT design is summarized in Figure 10.2. Authenticity of messages was assured by computing a message authentication code (MAC) at the sending bank and checking it at the receiving bank. The keys for this MAC used to be managed end-to-end: whenever a bank set up a relationship overseas, the senior manager who negotiated it would exchange keys with his opposite number, whether in a face-to-face meeting or afterwards by post to each others' home addresses. There would typically be two key components

to minimize the risk of compromise, with one sent in each direction (since even if a bank manager's mail is read in his mailbox by a criminal at one end, it's not likely to happen at both). The key was not enabled until both banks confirmed that it had been safely received and installed.

This way, SWIFT had no part in the message authentication; so long as the authentication algorithm in use was sound, none of their staff could forge a transaction. The authentication algorithm used is supposed to be a trade secret; but as banks like their security mechanisms to be international standards, a natural place to look might be the algorithm described in ISO 8731 [1094]. In this way, they got the worst of all possible worlds: the algorithm was fielded without the benefit of public analysis but got it later once it was expensive to change! An attack was found on the ISO 8731 message authentication algorithm and published in [1040] but, fortunately for the industry, it takes over 100,000 messages to recover a key — which is too large for a practical attack on a typical system that is used prudently.

Although SWIFT itself was largely outside the trust perimeter for message authentication, it did provide a non-repudiation service. Banks in each country sent their messages to a *Regional General Processor* (RGP) which logged them and forwarded them to SWIFT, which also logged them and sent them on to the recipient via the RGP in his country, which also logged them. The RGPs were generally run by different facilities management firms. Thus a bank (or a crooked bank employee) wishing to dishonestly repudiate a done transaction would have to subvert not just the local SWIFT application and its surrounding controls, but also two independent contractors in different countries. Note that the repudiation control from multiple logging is better than the integrity control. A bent bank wanting to claim that a transaction had been done when it hadn't could always try to insert the message between the other bank and their RGP; while a bent bank employee would probably just insert a bogus incoming message directly into a local system. So logs can be a powerful way of making repudiation difficult, and are much easier for judges to understand than cryptography.

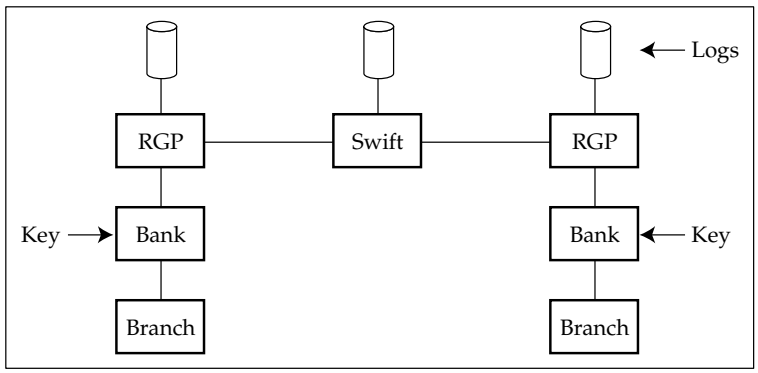

Figure 10.2: Architecture of SWIFT

Confidentiality depended on line encryption devices between the banks and the RGP node, and between these nodes and the main SWIFT processing sites. Key management was straightforward at first. Keys were hand carried in EEPROM cartridges between the devices at either end of a leased line. In countries where confidentiality was illegal, these devices could be omitted without impairing the authenticity and non-repudiation mechanisms.

Dual control was provided either by the use of specialized terminals (in small banks) or by mainframe software packages which could be integrated with a bank's main production system. The usual method of operation is to have three separate staff to do a SWIFT transaction: one to enter it, one to check it, and one to authorize it. (As the checker can modify any aspect of the message, this really only gives dual control, not triple control — and the programmers who maintain the interface can always attack the system there). Reconciliation was provided by checking transactions against daily statements received electronically from correspondent banks. This meant that someone who managed to get a bogus message into the system would sound an alarm within two or three days.

10.3.2 What Goes Wrong

SWIFT I ran for twenty years without a single report of external fraud. In the mid 1990s, it was enhanced by adding public key mechanisms; MAC keys are now shared between correspondent banks using public key cryptography and the MACs themselves may be further protected by a digital signature. The key management mechanisms have been ensconced as ISO standard 11166, which in turn has been used in other systems (such as CREST, which is used by banks and stockbrokers to register and transfer UK stocks and shares). There has been some debate over the security of this architecture [73, 1094]. Quite apart from the centralization of trust brought about by the adoption of public key cryptography — in that the central certification authority can falsely certify a key as belonging to a bank when it doesn't — CREST adopted 512-bit public keys, and these are too short: as I mentioned in the chapter on cryptology, at least one RSA public key of this length has been factored surreptitiously by a group of students [24].

However the main practical attacks on such systems have not involved the payment mechanisms themselves. The typical attack comes from a bank programmer inserting a bogus message into the processing queue. It usually fails because he does not understand the other controls in the system, or the procedural controls surrounding large transfers. For example, banks maintain accounts with each other, so when bank A sends money to a customer of bank B it actually sends an instruction 'please pay this customer the following sum out of our account with you'. As these accounts have both balances and credit limits, large payments aren't processed entirely automatically but need intervention from the dealing room to ensure that the needed currency or

credit line is made available. So transfers over a million dollars or so tend to need managerial interventions of which technical staff are ignorant; and there are also filters that look for large transactions so that the bank can report them to the money-laundering authorities if need be. There is also the common-sense factor, in that anyone who opens a bank account, receives a large incoming wire transfer and then starts frantically moving money out again used to need a very convincing excuse. (Common sense has become less of a backstop since 9/11 as customer due diligence and anti-money-laundering rules have become both formalised and onerous; bank staff rely more on box-ticking, which has made life easier for the bad guys [55].) In any case, the programmer who inserts a bogus transaction into the system usually gets arrested when he turns up to collect the cash. If your life's goal is a career in bank fraud, you're better off getting an accounting or law degree and working in a loans office rather than messing about with computers.

Other possible technical attacks, such as inserting Trojan software into the PCs used by bank managers to initiate transactions, wiretapping the link from the branch to the bank mainframe, subverting the authentication protocol used by bank managers to log on, and even inserting a bogus transaction in the branch LAN causing it to appear on the relevant printer — would also run up against the business-process controls. In fact, most large scale bank frauds which 'worked' have not used technical attacks but exploited procedural vulnerabilities.

- The classic example is a letter of guarantee. It is common enough for a company in one country to guarantee a loan to a company in another. This can be set up as a SWIFT message, or even a paper letter. But as no cash changes hands at the time, the balancing controls are inoperative. If a forged guarantee is accepted as genuine, the 'beneficiary' can take his time borrowing money from the accepting bank, laundering it, and disappearing. Only when the victim bank realises than the loan has gone sour and tries to call in the guarantee is the forgery discovered.
- An interesting fraud of a slightly different type took place in 1986 between London and Johannesburg. At that time, the South African government operated two exchange rates, and in one bank the manager responsible for deciding which rate applied to each transaction conspired with a rich man in London. They sent money out to Johannesburg at an exchange rate of seven Rand to the Pound, and back again the following day at four. After two weeks of this, the authorities became suspicious, and the police came round. On seeing them in the dealing room, the manager fled without stopping to collect his jacket, drove over the border to Swaziland, and flew via Nairobi to London. There, he boasted to the press about how he had defrauded the wicked apartheid system. As the UK has no exchange control, exchange control

fraud isn't an offence and so he couldn't be extradited. The conspirators got away with millions, and the bank couldn't even sue them.

- Perhaps the best known money transfer fraud occurred in 1979 when Stanley Rifkin, a computer consultant, embezzled over ten million dollars from Security Pacific National Bank. He got round the money laundering controls by agreeing to buy a large shipment of diamonds from a Russian government agency in Switzerland. He got the transfer into the system by observing an authorization code used internally when dictating transfers to the wire transfer department, and simply used it over the telephone (a classic example of dual control breakdown at a system interface). He even gave himself extra time to escape by doing the deal just before a U.S. bank holiday. Where he went wrong was in not planning what to do after he collected the stones. If he'd hid them in Europe, gone back to the USA and helped investigate the fraud, he might well have got away with it; as it was, he ended up on the run and got caught.

The system design lesson is unchanged: one must always be alert to things which defeat the dual control philosophy. However, as time goes on we have to see it in a broader context. Even if we can solve the technical problems of systems administration, interfaces and so on, there's still the business process problem of what we control — quite often critical transactions don't appear as such at a casual inspection.

10.4 Automatic Teller Machines

Another set of lessons about the difficulties and limitations of dual control emerges from studying the security of *automatic teller machines* (ATMs). ATMs, also known as cash machines, have been one of the most influential technological innovations of the 20th century.

ATMs were the first large-scale retail transaction processing systems. They were devised in 1938 by the inventor Luther Simjian, who also thought up the teleprompter and the self-focussing camera. He persuaded Citicorp to install his 'Bankamat' machine in New York in 1939; they withdrew it after six months, saying 'the only people using the machines were a small number of prostitutes and gamblers who didn't want to deal with tellers face to face' [1168]. Its commercial introduction dates to 1967, when a machine made by De La Rue was installed by Barclays Bank in Enfield, London. The world installed base is now thought to be about 1,500,000 machines. The technology developed for them is now also used in card payment terminals in shops. Modern block ciphers were first used on a large scale in ATM networks: they are used to generate and verify PINs in secure hardware devices located within the ATMs and at bank computer centres. This technology, including

block ciphers, tamper-resistant hardware and the supporting protocols, ended up being used in many other applications from postal franking machines to lottery ticket terminals. In short, ATMs were the 'killer app' that got modern commercial cryptology and retail payment technology off the ground.

10.4.1 ATM Basics

Most ATMs operate using some variant of a system developed by IBM for its 3624 series cash machines in the late 1970s. The card contains the customer's primary account number, *PAN*. A secret key, called the 'PIN key', is used to encrypt the account number, the decimalize it and truncate it. The result of this operation is called the 'natural PIN'; an offset can be added to it in order to give the PIN which the customer must enter. The offset has no real cryptographic function; it just enables customers to choose their own PIN. An example of the process is shown in Figure 10.3.

In the first ATMs to use PINs, each ATM contained a copy of the PIN key and each card contained the offset as well as the primary account number. Each ATM could thus verify all customer PINs. Early ATMs also operated offline; if your cash withdrawal limit was $500 per week, a counter was kept on the card. In recent years networks have become more dependable and ATMs have tended to operate online only, which simplifies the design; the cash counters and offsets have vanished from magnetic strips and are now kept on servers. In the last few years, magnetic strips have been supplemented with smartcard chips in some countries, especially in Europe; I will describe the smartcard systems later. However the basic principle remains: PINs are generated and protected using cryptography.

Dual control is implemented in this system using tamper-resistant hardware. A cryptographic processor, also known as a *security module*, is kept in the bank's server room and will perform a number of defined operations on customer PINs and on related keys in ways that enforce a dual-control policy. This includes the following.

1. Operations on the clear values of customer PINs, and on the keys needed to compute them or used to protect them, are all done in tamper-resistant

Account number *PAN*:	`8807012345691715`
PIN key *KP*:	`FEFEFEFEFEFEFEFE`
Result of DES $\{PAN\}_{KP}$:	`A2CE126C69AEC82D`
$\{N\}_{KP}$ decimalized:	`0224126269042823`
Natural PIN:	`0224`
Offset:	`6565`
Customer PIN:	`6789`

Figure 10.3: IBM method for generating bank card PINs

hardware and the clear values are never made available to any single member of the bank's staff.

2. Thus, for example, the cards and PINs are sent to the customer via separate channels. The cards are personalized in a facility with embossing and mag strip printing machinery, and the PIN mailers are printed in a separate facility containing a printer attached to a security module.
3. A *terminal master key* is supplied to each ATM in the form of two printed components, which are carried to the branch by two separate officials, input at the ATM keyboard, and combined to form the key. Similar procedures (but with three officials) are used to set up master keys between banks and network switches such as VISA.
4. If ATMs are to perform PIN verification locally, then the PIN key is encrypted under the terminal master key and then sent to the ATM.
5. If the PIN verification is to be done centrally over the network, the PIN is encrypted under a key set up using the terminal master key. It will then be sent from the ATM to a central security module for checking.
6. If the bank's ATMs are to accept other banks' cards, then its security modules use transactions that take a PIN encrypted under an ATM key, decrypt it and re-encrypt it for its destination, such as using a key shared with VISA. This *PIN translation* function is done entirely within the hardware security module, so that clear values of PINs are never available to the bank's programmers. VISA will similarly decrypt the PIN and re-encrypt it using a key shared with the bank that issued the card, so that the PIN can be verified by the security module that knows the relevant PIN key.

During the 1980s and 1990s, hardware security modules became more and more complex, as ever more functionality got added to support more complex financial applications from online transactions to smartcards. An example of a leading product is the IBM 4758 — this also has the virtue of having its documentation available publicly online for study (see [641] for the command set and [1195] for the architecture and hardware design). We'll discuss this later in the chapter on tamper resistance.

But extending the dual control security policy from a single bank to tens of thousands of banks worldwide, as modern ATM networks do, was not completely straightforward.

- When people started building ATM networks in the mid 1980s, many banks used software encryption rather than hardware security modules to support ATMs. So in theory, any bank's programmers might get access to the PINs of any other bank's customers. The remedy was to push through standards for security module use. In many countries

(such as the USA), these standards were largely ignored; but even where they were respected, some banks continued using software for transactions involving their own customers. So some keys (such as those used to communicate with ATMs) had to be available in software too, and knowledge of these keys could be used to compromise the PINs of other banks' customers. (I'll explain this in more detail later.) So the protection given by the hardware TCB was rarely complete.

- It is not feasible for 10,000 banks to share keys in pairs, so each bank connects to a switch provided by an organization such as VISA or Cirrus, and the security modules in these switches translate the traffic. The switches also do accounting, and enable banks to settle their accounts for each day's transactions with all the other banks in the system by means of a single electronic debit or credit. The switch is highly trusted, and if something goes wrong there the consequences can be severe. In one case, there turned out to be not just security problems but also dishonest staff. The switch manager ended up a fugitive from justice, and the bill for remediation was in the millions. In another case, a Y2K-related software upgrade at a switch was bungled, with the result that cardholders in one country found that for a day or two they could withdraw money even if their accounts were empty. This also led to a very large bill.
- Corners are cut to reduce the cost of dealing with huge transaction volumes. For example, it is common for authentication of authorization responses to be turned off. So anyone with access to the network can cause a given ATM to accept any card presented to it, by simply replaying a positive authorization response. Network managers claim that should a fraud ever start, then the authentication can always be turned back on. This might seem reasonable; attacks involving manipulated authorization responses are very rare. Similarly, after UK banks put smartcard chips into bank cards, some of them kept in accepting magnetic-strip transactions, so that a card with a broken chip would still work so long as the magnetic strip could be read. But such shortcuts — even when apparently reasonable on grounds of risk and cost — mean that a bank which stonewalls customer complaints by saying its ATM network is secure, and so the transaction must be the customer's fault, is not telling the truth. This may lay the bank and its directors open to fraud charges. What's more, changing the network's modus operandi suddenly in response to a fraud can be difficult; it can unmask serious dependability problems or lead to unacceptable congestion. This brings home the late Roger Needham's saying that 'optimization is the process of taking something which works, and replacing it by something which doesn't quite but is cheaper'.

There are many other ways in which ATM networks can be attacked in theory, and I'll discuss a number of them later in the context of interface security: the design of the hardware security modules that were in use for decades was so poor that programmers could extract PINs and keys simply by issuing suitable sequences of commands to the device's interface with the server, without having to break either the cryptographic algorithms or the hardware tamper-resistance. However, one of the interesting things about these systems is that they have now been around long enough, and have been attacked enough by both insiders and outsiders, to give us a lot of data points on how such systems fail in practice.

10.4.2 What Goes Wrong

ATM fraud is an interesting study as this is a mature system with huge volumes and a wide diversity of operators. There have been successive waves of ATM fraud, which have been significant since the early 1990s. In each wave, a set of vulnerabilities was exploited and then eventually fixed; but the rapidly growing scale of payment card operations opened up new vulnerabilities. There is a fascinating interplay between the technical and regulatory aspects of protection.

The first large wave of fraud lasted from perhaps 1990–96 and exploited the poor implementation and management of early systems. In the UK, one prolific fraudster, Andrew Stone, was convicted three times of ATM fraud, the last time getting five and a half years in prison. He got involved in fraud when he discovered by chance the 'encryption replacement' trick I discussed in the chapter on protocols: he changed the account number on his bank card to his wife's and found by chance that he could take money out of her account using his PIN. In fact, he could take money out of any account at that bank using his PIN. This happened because his bank (and at least two others) wrote the encrypted PIN to the card's magnetic strip without linking it to the account number in any robust way (for example, by using the 'offset' method described above). His second method was 'shoulder surfing': he'd stand in line behind a victim, observe the entered PIN, and pick up the discarded ATM slip. Most banks at the time printed the full account number on the slip, and a card would work with no other correct information on it.

Stone's methods spread via people he trained as his accomplices, and via a 'Howto' manual he wrote in prison. Some two thousand victims of his (and other) frauds banded together to bring a class action against thirteen banks to get their money back; the banks beat this on the technical legal argument that the facts in each case were different. I was an expert in this case, and used it write a survey of what went wrong [33] (there is further material in [34]). The fraud spread to the Netherlands, Italy and eventually worldwide, as criminals

learned a number of simple hacks. Here I'll summarize the more important and interesting lessons we learned.

The engineers who designed ATM security systems in the 1970s and 1980s (of whom I was one) had assumed that criminals would be relatively sophisticated, fairly well-informed about the system design, and rational in their choice of attack methods. In addition to worrying about the many banks which were slow to buy security modules and implementation loopholes such as omitting authentication codes on authorization responses, we agonized over whether the encryption algorithms were strong enough, whether the tamper-resistant boxes were tamper-resistant enough, and whether the random number generators used to manufacture keys were random enough. We knew we just couldn't enforce dual control properly: bank managers considered it beneath their dignity to touch a keyboard, so rather than entering the ATM master key components themselves after a maintenance visit, most of them would just give both key components to the ATM engineer. We wondered whether a repairman would get his hands on a bank's PIN key, forge cards in industrial quantities, close down the whole system, and wreck public confidence in electronic banking.

The great bulk of the actual 'phantom withdrawals', however, appeared to have one of the following three causes:

- Simple processing errors account for a lot of disputes. With U.S. customers making something like 5 billion ATM withdrawals a year, even a system that only makes one error per hundred thousand transactions will give rise to 50,000 disputes a year. In practice the error rate seems to lie somewhere between 1 in 10,000 and 1 in 100,000. One source of errors we tracked down was that a large bank's ATMs would send a transaction again if the network went down before a confirmation message was received from the mainframe; periodically, the mainframe itself crashed and forgot about open transactions. We also found customers whose accounts were debited with other customers' transactions, and other customers who were never debited at all for their card transactions. (We used to call these cards 'directors' cards' and joked that they were issued to bank directors.)
- Thefts from the mail were also huge. They are reckoned to account for 30% of all UK payment card losses, but most banks' postal control procedures have always been dismal. For example, when I moved to Cambridge in February 1992 I asked my bank for an increased card limit: the bank sent not one, but two, cards and PINs through the post. These cards arrived only a few days after intruders had got hold of our apartment block's mail and torn it up looking for valuables. It turned out that this bank did not have the systems to deliver a card by registered post. (I'd asked them to send the card to the branch for me to collect but the

branch staff had simply re-addressed the envelope to me.) Many banks now make you phone a call center to activate a card before you can use it. This made a dent in the fraud rates.

- Frauds by bank staff appeared to be the third big cause of phantoms. We mentioned the Hastings case in section 10.2.3 above; there are many others. For example, in Paisley, Scotland, an ATM repairman installed a portable computer inside an ATM to record customer card and PIN data and then went on a spending spree with forged cards. In London, England, a bank stupidly used the same cryptographic keys in its live and test systems; maintenance staff found out that they could work out customer PINs using their test equipment, and started offering this as a service to local criminals at £50 a card. Insider frauds were particularly common in countries like Britain where the law generally made the customer pay for fraud, and rarer in countries like the USA where the bank paid; British bank staff knew that customer complaints wouldn't be investigated carefully, so they got lazy, careless, and sometimes bent.

These failures are all very much simpler and more straightforward than the ones we'd worried about. In fact, the only fraud we had worried about and that actually happened to any great extent was on offline processing. In the 1980s, many ATMs would process transactions while the network was down, so as to give 24-hour service; criminals — especially in Italy and later in England too — learned to open bank accounts, duplicate the cards and then 'jackpot' a lot of ATMs overnight when the network was down [775]. This forced most ATM operations to be online-only by 1994.

However, there were plenty of frauds that happened in quite unexpected ways. I already mentioned the Utrecht case in section 3.5, where a tap on a garage point-of-sale terminal was used to harvest card and PIN data; and Stone's 'encryption replacement' trick. There were plenty more.

- Stone's shoulder-surfing trick of standing in an ATM queue, observing a customer's PIN, picking up the discarded ticket and copying the data to a blank card, was not in fact invented by him. It was first reported in New York in the mid 1980s; and it was still working in the Bay Area in the mid 1990s. By then it had been automated; Stone (and Bay area criminals) used video cameras with motion sensors to snoop on PINs, whether by renting an apartment overlooking an ATM or even parking a rented van there. Visual copying is easy to stop: the standard technique nowadays is to print only the last four digits of the account number on the ticket, and there's also a three-digit 'card verification value' (CVV) on the magnetic strip that should never be printed. Thus even if the villain's camera is good enough to read the account number and expiry date from the front of the card, a working copy can't be made. (The CVV

is like the three-digit security code on the signature strip, but different digits; each is computed from the account number and expiry date by encrypting them with a suitable key). Surprisingly, it still happens; I have a letter from a UK bank dated May 2007 claiming that a series of terrorist-related ATM frauds were perpetrated using closed-circuit TV. This amounts to an admission that the CVV is not always checked.

- There were some losses due to programming errors by banks. One small institution issued the same PIN to all its customers; another bank's cash machines had the feature that when a telephone card was entered at an ATM, it believed that the previous card had been inserted again. Crooks stood in line, observed customers' PINs, and helped themselves.
- There were losses due to design errors by ATM vendors. One model, common in the 1980s, would output ten banknotes from the lowest denomination non-empty cash drawer, whenever a certain fourteen digit sequence was entered at the keyboard. One bank printed this sequence in its branch manual, and three years later there was a sudden spate of losses. All the banks using the machine had to rush out a patch to disable the test transaction. And despite the fact that I documented this in 1993, and again in the first edition of this book in 2001, similar incidents are still reported in 2007. Some makes of ATM used in stores can be reprogrammed into thinking that they are dispensing $1 bills when in fact they're dispensing twenties; it just takes a default master password that is printed in widely-available online manuals. Any passer-by who knows this can stroll up to the machine, reset the bill value, withdraw $400, and have his account debited with $20. The store owners who lease the machines are not told of the vulnerability, and are left to pick up the tab [1037].
- Several banks thought up check-digit schemes to enable PINs to be checked by offline ATMs without having to give them the bank's PIN key. For example, customers of one British bank get a credit card PIN with digit one plus digit four equal to digit two plus digit three, and a debit card PIN with one plus three equals two plus four. Crooks found they could use stolen cards in offline devices by entering a PIN such as 4455.
- Many banks' operational security procedures were simply dire. In August 1993, my wife went into a branch of our bank with a witness and told them she'd forgotten her PIN. The teller helpfully printed her a new PIN mailer from a printer attached to a PC behind the counter — just like that! It was not the branch where our account is kept. Nobody knew her and all the identification she offered was our bank card and her checkbook. When anyone who's snatched a handbag can walk in off

the street and get a PIN for the card in it at any branch, no amount of encryption technology will do much good. (The bank in question has since fallen victim to a takeover.)

- A rapidly growing modus operandi in the early 1990s was to use false terminals to collect card and PIN data. The first report was from the USA in 1988; there, crooks built a vending machine which would accept any card and PIN, and dispense a packet of cigarettes. In 1993, two villains installed a bogus ATM in the Buckland Hills Mall in Connecticut [667, 962]. They had managed to get a proper ATM and a software development kit for it — all bought on credit. Unfortunately for them, they decided to use the forged cards in New York, where cash machines have hidden video cameras, and as they'd crossed a state line they ended up getting long stretches in Club Fed.

So the first thing we did wrong when designing ATM security systems in the early to mid 1980s was to worry about criminals being clever, when we should rather have worried about our customers — the banks' system designers, implementers and testers — being stupid. Crypto is usually only part of a very much larger system. It gets a lot of attention because it's mathematically interesting; but as correspondingly little attention is paid to the 'boring' bits such as training, usability, standards and audit, it's rare that the bad guys have to break the crypto to compromise a system. It's also worth bearing in mind that there are so many users for large systems such as ATM networks that we must expect the chance discovery and exploitation of accidental vulnerabilities which were simply too obscure to be caught in testing.

The second thing we did wrong was to not figure out what attacks could be industrialised, and focus on those. In the case of ATMs, the false-terminal attack is the one that made the big time. The first hint of organised crime involvement was in 1999 in Canada, where dozens of alleged Eastern European organized-crime figures were arrested in the Toronto area for deploying doctored point-of-sale terminals [85, 152]. The technology has since become much more sophisticated; 'skimmers' made in Eastern Europe are attached to the throats of cash machines to read the magnetic strip and also capture the PIN using a tiny camera. I'll discuss these in more detail in the next section. Despite attempts to deal with false-terminal attacks by moving from magnetic strip cards to smartcards, they have become pervasive. They will be difficult and expensive to eliminate.

10.4.3 Incentives and Injustices

In the USA, the banks have to carry the risks associated with new technology. This was decided in a historic precedent, Judd versus Citibank, in which bank customer Dorothy Judd claimed that she had not made some disputed withdrawals and Citibank said that as its systems were secure, she must have done.

The judge ruled that Citibank's claim to infallibility was wrong in law, as it put an unmeetable burden of proof on her, and gave her her money back [674]. The U.S. Federal Reserve incorporated this into 'Regulation E', which requires banks to refund all disputed transactions unless they can prove fraud by the customer [440]. This has led to some minor abuse — misrepresentations by customers are estimated to cost the average U.S. bank about $15,000 a year — but this is an acceptable cost (especially as losses from vandalism are typically three times as much) [1362].

In other countries — such as the UK, Germany, the Netherlands and Norway — the banks got away for many years with claiming that their ATM systems were infallible. Phantom withdrawals, they maintained, could not possibly exist and a customer who complained of one must be mistaken or lying. This position was demolished in the UK when Stone and a number of others started being jailed for ATM fraud, as the problem couldn't be denied any more. Until that happened, however, there were some rather unpleasant incidents which got banks a lot of bad publicity [34]. The worst was maybe the Munden case.

John Munden was one of our local police constables, based in Bottisham, Cambridgeshire; his beat included the village of Lode where I lived at the time. He came home from holiday in September 1992 to find his bank account empty. He asked for a statement, found six withdrawals for a total of £460 which he did not recall making, and complained. His bank responded by having him prosecuted for attempting to obtain money by deception. It came out during the trial that the bank's system had been implemented and managed in a ramshackle way; the disputed transactions had not been properly investigated; and all sorts of wild claims were made by the bank, such as that their ATM system couldn't suffer from bugs as its software was written in assembler. Nonetheless, it was his word against the bank's. He was convicted in February 1994 and sacked from the police force.

This miscarriage of justice was overturned on appeal, and in an interesting way. Just before the appeal was due to be heard, the prosecution served up a fat report from the bank's auditors claiming that the system was secure. The defense demanded equal access to the bank's systems for its own expert. The bank refused and the court therefore disallowed all its computer evidence — including even its bank statements. The appeal succeeded, and John got reinstated. But this was only in July 1996 — he'd spent the best part of four years in limbo and his family had suffered terrible stress. Had the incident happened in California, he could have won enormous punitive damages — a point bankers should ponder as their systems become global and their customers can be anywhere.[2]

[2]Recently the same drama played itself out again when Jane Badger, of Burton-on-Trent, England, was prosecuted for complaining about phantom withdrawals. The case against her collapsed in January 2008. The bank, which is called Egg, is a subsidiary of Citicorp.

The lesson to be drawn from such cases is that dual control is not enough. If a system is to provide evidence, then it must be able to withstand examination by hostile experts. In effect, the bank had used the wrong security policy. What they really needed wasn't dual control but *non-repudiation*: the ability for the principals in a transaction to prove afterwards what happened. This could have been provided by installing ATM cameras; although these were available (and are used in some U.S. states), they were not used in Britain. Indeed, during the 1992–4 wave of ATM frauds, the few banks who had installed ATM cameras were pressured by the other banks into withdrawing them; camera evidence could have undermined the stance that the banks took in the class action that their systems were infallible.

One curious thing that emerged from this whole episode was that although U.S. banks faced a much fiercer liability regime, they actually spent less on security that UK banks did, and UK banks suffered more fraud. This appears to have been a moral-hazard effect, and was one of the anomalies that sparked interest in security economics. Secure systems need properly aligned incentives.

10.5 Credit Cards

The second theme in consumer payment systems is the credit card. For many years after their invention in the 1950s, credit cards were treated by most banks as a loss leader with which to attract high-value customers. Eventually, in most countries, the number of merchants and cardholders reached critical mass and the transaction volume suddenly took off. In Britain, it took almost twenty years before most banks found the business profitable; then all of a sudden it was extremely profitable. Payment systems have strong network externalities, just like communications technologies or computer platforms: they are two-sided markets in which the service provider must recruit enough merchants to appeal to cardholders, and vice versa. Because of this, and the huge investment involved in rolling out a new payment system to tens of thousands of banks, millions of merchants and billions of customers worldwide, any new payment mechanism is likely to take some time to get established. (The potentially interesting exceptions are where payment is bundled with some other service, such as with Google Checkout.)

Anyway, when you use a credit card to pay for a purchase in a store, the transaction flows from the merchant to his bank (the acquiring bank) which pays him after deducting a *merchant discount* of typically 4–5%. If the card was issued by a different bank, the transaction now flows to a switching center run by the brand (such as VISA) which takes a commission and passes it to the issuing bank for payment. Daily payments between the banks and the brands settle the net cash flows. The issuer also gets a slice of the merchant discount, but makes most of its money from extending credit to cardholders at rates that are usually much higher than the interbank rate.

10.5.1 Fraud

The risk of fraud using stolen cards was traditionally managed by a system of *hot card lists* and merchant *floor limits*. Each merchant gets a local hot card list — formerly on paper, now stored in his terminal — plus a limit set by their acquiring bank above which they have to call for authorization. The call center, or online service, which he uses for this has access to a national hot card list; above a higher limit, they will contact VISA or MasterCard which has a complete list of all hot cards being used internationally; and above a still higher limit, the transaction will be checked all the way back to the card issuer. Recently, the falling cost of communications has led to many transactions being authorised all the way back to the issuer, but there are still extensive fallback processing capabilities. This is because maintaining 99.9999% availability on a network, plus the capacity to handle peak transaction volumes on the Wednesday before Thanksgiving and the Saturday just before Christmas, still costs a whole lot more than the fraud from occasional offline and stand-in processing.

The introduction of *mail order and telephone order* (MOTO) transactions in the 1970s meant that the merchant did not have the customer present, and was not able to inspect the card. What was to stop someone ordering goods using a credit card number he'd picked up from a discarded receipt?

Banks managed the risk by using the expiry date as a password, lowering the floor limits, increasing the merchant discount and insisting on delivery to a cardholder address, which is supposed to be checked during authorization. But the main change was to shift liability so that the merchant bore the full risk of disputes. If you challenge an online credit card transaction (or in fact any transaction made under MOTO rules) then the full amount is immediately debited back to the merchant, together with a significant handling fee. The same procedure applies whether the debit is a fraud, a dispute or a return.

A recent development has been the 'Verified by VISA' program under which merchants can refer online credit-card transactions directly to the issuing bank, which can then authenticate the cardholder using its preferred method. The incentive for the merchant is that the transaction is then treated as a cardholder-present one, so the merchant is no longer at risk. The problem with this is that the quality of authentication offered by participating banks varies wildly. At the top of the scale are banks that use two-channel authentication: when you buy online you get a text message saying something like 'If you really want to pay Amazon.com $76.23, enter the code 4697 in your browser now'. At the bottom end are banks that ask you to enter your ATM PIN into the browser directly — thereby making their customers wide-open targets for particularly severe phishing attacks. There is a clear disincentive for the cardholder, who may now be held liable in many countries regardless of the quality of the local authentication methods.

Of course, even if you have the cardholder physically present, this doesn't guarantee that fraud will be rare. For many years, most fraud was done in person with stolen cards, and stores which got badly hit tended to be those selling goods that can be easily fenced, such as jewelry and consumer electronics. Banks responded by lowering their floor limits. More recently, as technical protection mechanisms have improved, there has been an increase in scams involving cards that were never received by genuine customers. This *pre-issue fraud* can involve thefts from the mail of the many 'pre-approved' cards which arrive in junk mail, or even applications made in the names of people who exist and are creditworthy, but are not aware of the application ('identity theft'). These attacks on the system are intrinsically hard to tackle using purely technical means.

10.5.2 Forgery

In the early 1980's, electronic terminals were introduced through which a sales clerk could swipe a card and get an authorization automatically. But the sales draft was still captured from the embossing, so crooks figured out how to re-encode the magnetic strip of a stolen card with the account number and expiry date of a valid card, which they often got by fishing out discarded receipts from the trash cans of expensive restaurants. A re-encoded card would authorize perfectly, but when the merchant submitted the draft for payment, the account number didn't match the authorization code (a six digit number typically generated by encrypting the account number, date and amount). So the merchants didn't get paid and raised hell.

Banks then introduced *terminal draft capture* where a sales draft is printed automatically using the data on the card strip. The crooks' response was a flood of forged cards, many produced by Triad gangs: between 1989 and 1992, magnetic strip counterfeiting grew from an occasional nuisance into half the total fraud losses [7]. VISA's response was *card verification values* (CVVs) — these are three-digit MACs computed on the card strip contents (account number, version number, expiry date) and written at the end of the strip. They worked well initially; in the first quarter of 1994, VISA International's fraud losses dropped by 15.5% while Mastercard's rose 67% [269]. So Mastercard adopted similar checksums too.

The crooks moved to *skimming* — operating businesses where genuine customer cards were swiped through an extra, unauthorized, terminal to grab a copy of the magnetic strip, which would then be re-encoded on a genuine card. The banks' response was intrusion detection systems, which in the first instance tried to identify criminal businesses by correlating the previous purchase histories of customers who complained.

In the late 1990's, credit card fraud rose sharply due to another simple innovation in criminal technology: the operators of the crooked businesses

which skim card data absorb the cost of the customer's transaction rather than billing it. You have a meal at a Mafia-owned restaurant, offer a card, sign the voucher, and fail to notice when the charge doesn't appear on your bill. Perhaps a year later, there is suddenly a huge bill for jewelry, electrical goods or even casino chips. By then you've completely forgotten about the meal, and the bank never had a record of it [501].

In the early 2000's, high-tech criminals became better organised as electronic crime became specialised. Phishing involved malware writers, botnet herders, phishing site operators and cash-out specialists, linked by black markets organised in chat rooms. This has spilled over from targeting online transactions to attacks on retail terminals. Fake terminals, and terminal tapping devices, used in the USA and Canada simply record mag-strip card and PIN data, which are used to make card clones for use in ATMs. In the Far East, wiretaps have been used to harvest card data wholesale [792]. Things are more complex in Europe which has introduced smartcards, but there are now plenty of devices that copy the EMV standard smartcards to mag-strip cards that are used in terminals that accept mag-strip transactions. Some of them use vulnerabilities in the EMV protocol, and so I'll come back to discuss them after I've described bank smartcard use in the next section.

10.5.3 Automatic Fraud Detection

There has been a lot of work since the mid-1990s on more sophisticated financial intrusion detection. Some generic systems do abuse detection using techniques such as neural networks, but it's unclear how effective they are. When fraud is down one year, it's hailed as a success for the latest fraud spotting system [101], while when the figures are up a few years later the vendors let the matter pass quietly [1191].

More convincing are projects undertaken by specific store chains that look for known patterns of misuse. For example, an electrical goods chain in the New York area observed that offender profiling (by age, sex, race and so on) was ineffective, and used purchase profiling instead to cut fraud by 82% in a year. Their technique involved not just being suspicious of high value purchases, but training staff to be careful when customers were careless about purchases and spent less than the usual amount of time discussing options and features. These factors can be monitored online too, but one important aspect of the New York success is harder for a web site: employee rewarding. Banks give a $50 reward per bad card captured, which many stores just keep — so their employees won't make an effort to spot cards or risk embarrassment by confronting a customer. In New York, some store staff were regularly earning a weekly bonus of $150 or more [840].

With online shopping, the only psychology the site designer can leverage is that of the villain. It has been suggested that an e-commerce site should have

an unreasonably expensive 'platinum' option that few genuine customers will want to buy. This performs two functions. First, it helps you to do basic purchase profiling. Second, it fits with the model of *Goldilocks pricing* developed by economists Hal Shapiro and Carl Varian, who point out that the real effect of airlines' offering first class fares is to boost sales of business class seats to travelers who can now convince their bosses (or themselves) that they are being 'economical' [1159]. Another idea is to have a carefully engineered response to suspect transactions: if you just say 'bad card, try another one' then the fraudster probably will. You may even end up being used by the crooks as an online service that tells them which of their stolen cards are on the hot list, and this can upset your bank (even though the banks are to blame for the system design). A better approach is claim that you're out of stock, so the bad guy will go elsewhere [1199].

As for electronic banking, it has recently become important to make intrusion detection systems work better with lower-level mechanisms. A good example is the real-time man-in-the-middle attack. After banks in the Netherlands handed out password calculators to their online banking customers, the response of the phishermen was to phish in real time: the mark would log on to the phishing site, which would then log on to the bank site and relay the challenge for the mark to enter into his calculator. The quick fix for this is to look out for large numbers of logons coming from the same IP address. It's likely to be a long struggle, of course; by next year, the bad guys may be using botnets to host the middleman software.

10.5.4 The Economics of Fraud

There's a lot of misinformation about credit card fraud, with statistics quoted selectively to make points. In one beautiful example, VISA was reported to have claimed that card fraud was up, and that card fraud was down, on the same day [598].

But a consistent pattern of figures can be dug out of the trade publications. The actual cost of credit card fraud during the 1990s was about 0.15% of all international transactions processed by VISA and Mastercard [1087], while national rates varied from 1% in America through 0.2% in UK to under 0.1% in France and Spain. The prevailing business culture has a large effect on the rate. U.S. banks, for example, are much more willing to send out huge junk mailings of pre-approved cards to increase their customer base, and write off the inevitable pre-issue fraud as a cost of doing business. In other countries, banks are more risk-averse.

The case of France is interesting, as it seems at first sight to be an exceptional case in which a particular technology has brought real benefits. French banks introduced chip cards for all domestic transactions in the late 1980's, and this reduced losses from 0.269% of turnover in 1987 to 0.04% in 1993 and 0.028%

in 1995. However, there is now an increasing amount of cross-border fraud. French villains use foreign magnetic stripe cards — particularly from the UK [498, 1087] — while French chip cards are used at merchants in non-chip countries [270]. But the biggest reduction in Europe was not in France but in Spain, where the policy was to reduce all merchant floor limits to zero and make all transactions online. This cut their losses from 0.21% of turnover in 1988 to 0.008% in 1991 [110].

The lessons appear to be that first, card fraud is cyclical as new defences are introduced and the villains learn to defeat them; and second, that the most complicated and expensive technological solution doesn't necessarily work best in the field. In fact, villains get smarter all the time. After the UK moved from magnetic strip cards to chipcards in 2005, it took less than eighteen months for the crooks to industrialise the process of moving stolen card data abroad: by 2007, as I'll discuss shortly.

10.5.5 Online Credit Card Fraud — the Hype and the Reality

Turning now from traditional credit card fraud to the online variety, I first helped the police investigate an online credit card fraud in 1987. In that case, the bad guy got a list of hot credit card numbers from his girlfriend who worked in a supermarket, and used them to buy software from companies in California, which he downloaded to order for his customers. This worked because hot card lists at the time carried only those cards which were being used fraudulently in that country; it also guaranteed that the bank would not be able to debit an innocent customer. As it happens, the criminal quit before there was enough evidence to nail him. A rainstorm washed away the riverbank opposite his house and exposed a hide which the police had built to stake him out.

From about 1995, there was great anxiety at the start of the dotcom boom that the use of credit cards on the Internet would lead to an avalanche of fraud, as 'evil hackers' intercepted emails and web forms and harvested credit card numbers by the million. These fears drove Microsoft and Netscape to introduce SSL/TLS to encrypt credit card transactions en route from browsers to web servers. (There was also a more secure protocol, SET, in which the browser would get a certificate from the card-issuing bank and would actually sign the transaction; this failed to take off as the designers didn't get the incentives right.)

The hype about risks to credit card numbers was overdone. Intercepting email is indeed possible but it's surprisingly difficult in practice — so much so that governments had to bully ISPs to install snooping devices on their networks to make court-authorized wiretaps easier [187]. I'll discuss this further in Part III. The actual threat is twofold. First, there's the growth of

phishing since 2004; there (as I remarked in Chapter 2) the issue is much more psychology than cryptography. TLS per se doesn't help, as the bad guys can also get certificates and encrypt the traffic.

Second, most of the credit card numbers that are traded online got into bad hands because someone hacked a merchant's computer. VISA had had rules for many years prohibiting merchants from storing credit card data once the transaction had been processed, but many merchants simply ignored them. From 2000, VISA added new rules that merchants had to install a firewall, keep security patches up-to-date, encrypt stored and transmitted data and regularly update antivirus software [1262]. These were also not enforced. The latest set of rules, the Payment Card Industry Data Security Standard, are a joint effort by VISA and Mastercard, and supported by the other brands too; they say much the same things, and we'll have to wait and see whether the enforcement is any better. 'PCI', as the new system's called, certainly seems to be causing some pain; in October 2007, the U.S. National Retail Federation asked credit card companies to stop forcing retailers to store credit card data at all (at present they are supposed to store card numbers temporarily in case of chargebacks) [1296].

The real incentives facing merchants are, first, the cost of disputes, and second, the security-breach disclosure laws that are (in 2007) in force in 34 U.S. states and that are contemplated as a European Directive. Disclosure laws have had a very definite effect in the USA as the stock prices of companies suffering a breach can fall several percent. As for disputes, consumer protection laws in many countries make it easy to repudiate a transaction. Basically all the customer has to do is call the credit card company and say 'I didn't authorize that' and the merchant is saddled with the bill. This was workable in the days when almost all credit card transactions took place locally and most were for significant amounts. If a customer fraudulently repudiated a transaction, the merchant would pursue them through the courts and harrass them using local credit reference agencies. In addition, the banks' systems are often quite capable of verifying local cardholder addresses.

But the Internet differs from the old mail order/telephone order regime in that many transactions are international, amounts are small, and verifying overseas addresses via the credit card system is problematic. Often all the call center operator can do is check that the merchant seems confident when reading an address in the right country. So the opportunity for repudiating transactions — and getting away with it — is hugely increased. There are particularly high rates of repudiation of payment to porn sites. No doubt some of these disputes happen when a transaction made under the influence of a flush of hormones turns up on the family credit card bill and the cardholder has to repudiate it to save his marriage; but many are the result of blatant fraud by operators. A common scam was to offer a 'free tour' of the site and demand a credit card number, supposedly to verify that the user was over 18,

and then bill him anyway. Some sites billed other consumers who have never visited them at all [620]. Even apparently large and 'respectable' web sites like `playboy.com` were criticised for such practices, and at the bottom end of the porn industry, things are atrocious.

The main brake on wicked websites is the credit-card chargeback. A bank will typically charge the merchant $100–200 in fees for each of them, as well as debiting the transaction amount from his account. So if more than a small percentage of the transactions on your are challenged by customers, your margins will be eroded. If chargebacks go over perhaps 10%, your bank may withdraw your card acquisition service. This has happened to a number of porn sites; a more prosaic example was the collapse of sportswear merchant `boo.com` because they had too many returns: their business model assumed a no-quibble exchange or refund policy, similar to those operated by high-street discount clothing stores. Yet more of their shipments than they'd expected were the wrong size, or the wrong colour, or just didn't appeal to the customers. Refunds are cheaper than chargebacks, but still, the credit card penalties broke the company [1199]. Chargebacks also motivate merchants to take case — to beware of odd orders (e.g. for four watches), orders from dodgy countries, customers using free email services, requests for expedited delivery, and so on. But leaving the bulk of the liability for mail-order transactions with them is suboptimal: the banks know much more about fraud patterns.

This history suggests that purely technological fixes may not be easy, and that the most effective controls will be at least partly procedural. Some card issuers offer credit card numbers that can be used once only; as they issue them one at a time to customers via their web site, this also helps drive lots of traffic to their advertisers [324]. Other banks have found that they get better results by investing in address verification [102]. However the big investment in the last few years has been in new card technologies, with Europe replacing both credit cards and debit cards with smartcards complying with the EMV 'chip and PIN' standard, while U.S. banks are starting to roll out bank cards based on RFID.

10.6 Smartcard-Based Banking

In the 1960s and 70s, various people proposed putting integrated circuits in bank cards. The Germans consider the smartcard to have been invented by Helmut Gröttrup and Jürgen Dethloff in 1968, when they proposed and patented putting a custom IC in a card 1968; the French credit Roland Moreno, who proposed putting memory chips in cards in 1973, and Michel Ugon who proposed adding a microprocessor in 1977. The French company Honeywell-Bull patented a chip containing memory, a microcontroller and everything else needed to do transactions in 1982; they started being used in French

pay phones in 1983; and in banking from the mid-1980s, as discussed in section 10.5.4 above.

Smartcards were marketed from the beginning as the French contribution to the information age, and the nascent industry got huge government subsidies. In the rest of the world, progress was slower. There were numerous pilot projects in which smartcards were tried out with different protocols, and in different applications. I already mentioned the COPAC system at 3.8.1; we developed this in 1991–2 for use in countries with poor telecommunications, and it sold best in Russia. Norway's commercial banks started issuing smartcards in 1986 but its savings banks refused to; when the central bank pressured the banks to unite on a common technology, mag stripe won and smartcards were withdrawn in 1995. Britain's NatWest Bank developed the Mondex electronic purse system in the early 90s, piloted it in Swindon, then sold it to Mastercard. There was a patent fight between VISA (which had bought the COPAC rights) and Mastercard. The Belgian banks implemented an electronic purse called 'Proton' for low-value payments to devices like parking meters; the Germans followed with 'Geldkarte' which became the European standard EN1546 and is now also available as the 'Moneo' electronic purse in France.

Offline systems such as Mondex had problems dealing with broken cards. If the back-end system doesn't do full balancing, then when a customer complains that a card has stopped working, all the bank can do is either to refund the amount the customer claims was on the card, or tell her to get lost; so most modern systems do balancing, which means they aren't as cheap to operate as one might have hoped. All this was good learning experience. But for a payment card to be truly useful, it has to work internationally — and especially so in Europe with many small countries jammed up close together, where even a one-hour shopping trip in the car may involve international travel. So the banks finally got together with their suppliers and hammered out a standard.

10.6.1 EMV

The EMV standards are named after the participating institutions Europay, Mastercard and VISA (Europay developed the Belgian Proton card). As of 2007, several hundred million European cardholders now have debit and credit cards that conform to this standard, and can be used more or less interoperably in the UK, Ireland, France, Germany and other participating countries. In English speaking countries such as the UK and Ireland, EMV has been branded as 'chip and PIN' (although the standards do also support signature-based transactions). The standards' proponents hope that they will become the worldwide norm for card payments, although this is not quite a done deal: Japan and increasingly the USA are adopting RFID standards for contactless payment, which I'll discuss in the next section. Anyway, in much

of the world, the EMV standards act as a 'fraud bulldozer', moving around the payment-systems landscape so that some types of fraud become less common and others more so.

The EMV protocol documents [429] are not so much a single protocol as a suite of protocols. The VISA version of the protocols alone come to more than 3,600 pages, and these are only the compatibility aspects — there are further documents specific to individual banks. Specifications this complex cannot be expected to be bug-free, and I'll describe some of the bugs in the following sections. The most obvious problem is that the documents allow many options, some of which are dangerous, either individually or in combination. So EMV can be thought of as a construction kit for building payment systems, with which one can build systems that are quite secure, or very insecure, depending on how various parameters are set, what sort of fallback modes are invoked on failure, and what other systems are hooked up.

In order to understand this, we need to look briefly at the EMV mechanisms. Each customer card contains a smartcard chip with the capability to verify a PIN and authenticate a transaction. The cards come in two types: low-cost cards that do only symmetric cryptography and use a set of protocols known as static data authentication (SDA); and more expensive cards that can generate digital signatures, supporting protocols called dynamic data authentication (DDA) and combined data authentication (CDA).

10.6.1.1 Static Data Authentication

SDA is the default EMV protocol, and it works as follows. The customer puts her card into the 'chip and PIN' terminal to which it sends a digital certificate, account number and the other data found on the old-fashioned magnetic strip, plus a digital signature from the card-issuing bank (the bank chooses which data items to sign). The terminal verifies the signature and the merchant enters the payment amount; the terminal solicits the PIN; the customer enters it; and it's sent in clear to the card. If the PIN is accepted, the card tells the terminal that all's well and generates a MAC, called an 'application data cryptogram', on the supplied data (merchant ID, amount, serial number, nonce and so on). The key used to compute this MAC is shared between the card and the customer's bank, and so it can only be verified by the issuing bank. (The bank could thus use any algorithm it liked, but the default is DES-CBC-MAC with triple-DES for the last block.) Also, the only way the terminal can check that the transaction is genuine is by going online and getting an acknowledgement. As this isn't always convenient, some merchants have a 'floor limit' below which offline transactions are permitted.

This protocol has a number of vulnerabilities that are by now well known. The most commonly-exploited one is backwards compatibility with magnetic strip cards: as the certificate contains all the information needed to forge a

mag-strip card, and as the introduction of chip and PIN means that people now enter PINs everywhere rather than just at ATMs, a number of gangs have used assorted sniffers to collect card data from terminals and collected money using mag-strip forgeries. Many ATMs and merchant terminals even in the EMV adopter countries will fall back to mag-strip processing for reliability reasons, and there are many countries — from the USA to Thailand — that haven't adopted EMV at all. There are two flavours of attack: where the PIN is harvested along with the card details, and where it's harvested separately.

First, where the card reader and the PIN pad are separate devices, then a wiretap between them will get PINs as well as card data. Since 2005 there have been reports of sniffing devices, made in Eastern Europe, that have been found in stores in Italy; they harvest the card and PIN data and send it by SMS to the villains who installed them. This may be done under cover of a false-pretext 'maintenance' visit or by corrupt store staff. There have also been reports of card cloning at petrol stations after PIN pads were replaced with tampered ones; although these cases are waiting to come to trial as I write, I investigated the tamper-resistance of PIN pads with two colleagues and we found that the leading makes were very easy to compromise.

For example, the Ingenico i3300, one of the most widely-deployed terminals in the UK in 2007, suffers from a series of design flaws. Its rear has a user-accessible compartment, shown in Figure 10.4, for the insertion of optional extra components. This space is not designed to be tamper-proof, and when covered it cannot be inspected by the cardholder even if she handles the device. This compartment gives access to the bottom layer of the circuit board. This does not give direct access to sensitive data — but, curiously, the designers opted to provide the attacker 1 mm diameter holes (used for positioning the optional components) and vias through the circuit board. From there, a simple metal hook can tap the serial data line. We found that a 1 mm diameter via, carrying the serial data signal, is easily accessed using a bent paperclip. This can be inserted through a hole in the plastic surrounding the internal compartment, and does not leave any external marks. The effect is that the attacker can design a small wiretap circuit that sits invisibly inside the terminal and gathers both card and PIN data. This circuit can be powered from the terminal itself and could contain a small mobile phone to SMS the booty to its makers.

Britain had an epidemic of fraud in 2006–7 apparently involving sniffer devices inserted into the wiring between card terminals and branch servers in petrol stations in the UK. As the card readers generally have integral PIN pads in this application, the PINs may be harvested by eye by petrol-station staff, many of whom are Tamils who arrived as refugees from the civil war in Sri Lanka. It's said that the Tamil Tigers — a terrorist group — intimidates them into participating. This was discovered when Thai police caught men in Phuket with 5,000 forged UK debit and credit cards, copied on to 'white plastic'.

Figure 10.4: A rigid wire is inserted through a hole in the Ingenico's concealed compartment wall to intercept the smartcard data. The front of the device is shown on the top right.

Attacks exploiting the fact that the MAC can't be read by the merchant include 'yescards'. These are cards programmed to accept any PIN (hence the name) and to participate in the EMV protocol using an externally-supplied certificate, returning random values for the MAC. A villain with a yescard and access to genuine card certificates — perhaps through a wiretap on a merchant terminal — can copy a cert to a yescard, take it to a merchant with a reasonable floor limit, and do a transaction using any PIN. This attack has been reported in France and suspected in the UK [122]; it's pushing France towards a move from SDA to DDA. However, most such frauds in Britain still use magnetic strip fallback: many ATMs and merchants use the strip if the chip is not working.

Another family of problems with EMV has to do with authentication methods. Each card, and each terminal, has a list of preferred methods, which might say in effect: 'first try online PIN verification, and if that's not supported use local cleartext PIN verification, and if that's not possible then you don't need to authenticate the customer at all'. It might at first sight be surprising that 'no authentication' is ever an option, but it's frequently there, in order to support devices such as parking ticket vending machines that don't have

PIN pads. One glitch is that the list of authentication methods isn't itself authenticated, so a bad man might manipulate it in a false-terminal or relay attack. Another possibility is to have two cards: your own card, for which you know the PIN, and a stolen card for which you don't, slotted into a device that lets you switch between them. You present the first card to the terminal and verify the PIN; you then present the transaction to the stolen card with the verification method changed to 'no authentication'. The stolen card computes the MAC and gets debited. The bank then maintains to the victim that as his chip was read and a PIN was used, he's liable for the money.

One countermeasure being contemplated is to insert the verification method into the transaction data; another is to reprogram cards to remove 'no authentication' from the list of acceptable options. If your bank takes the latter option, you'd better keep some change in your car ashtray! Yet another is to reprogram customer cards so that 'no authentication' works only up to some limit, say $200.

The fact that banks can now reprogram customers' cards in the field is also novel. The mechanism uses the shared key material and kicks in when the card's at an online terminal such as an ATM. One serious bug we discovered is that the encryption used to protect these messages is so poorly implemented that bank insiders can easily extract the keys from the hardware security modules [12]. I'll discuss this kind of vulnerability at greater length when we dive into the thicket of API security. Remote reprogrammability was pioneered by the pay-TV stations in their wars against pirates who cloned their smartcards; it can be a powerful tool, but it can also be badly misused. For example, it opens the possibility of a disgruntled insider launching a service-denial attack that rapidly wipes out all a bank's customers' cards.

However, such bankers' nightmares aside, the practical security of EMV depends to a great extent on implementation details such as the extent to which fallback magnetic-strip processing is available in local ATMs, the proportion of local shops open to various kinds skimmer attacks (whether because of personnel vulnerability factors or because there are many store chains using separate card readers and PIN pads), and — as always — incentives. Do the banks carry the can for fraud as in the USA, which makes them take care, or are they able to dump the costs on merchants and cardholders, as in much of Europe, which blunts their motivation to insist on high standards? Indeed, in some countries — notably the UK — banks appear to have seen EMV not so much as a fraud-reduction technology but a liability-engineering one. In the old days they generally paid for fraud in signature-based transactions but often blamed the customer for the PIN-based ones ('your PIN was used so you must have been negligent'). The attractions of changing most in-store transactions from signatures to PINs were obvious.

The bottom-line question is, of course, whether it paid for itself. In Britain it hasn't. Fraud rose initially, thanks to the much larger number of cards stolen

from the mail during the changeover period; local fraud has been said to fall since, though this has been achieved with the help of some fairly blatant manipulation of the numbers. For example, bank customers were stopped from reporting card fraud to the police from April 2007; frauds must be reported to the bank. Oh, and the banks have taken over much of the financing of the small police unit that does still investigate card fraud. This helps the government massage the crime statistics downward, and lets the banking industry control such prosecutions as do happen. Meanwhile overseas fraud has rocketed, thanks to the Tamil Tigers and to the vigorous international trade in stolen card numbers. The net effect was that by October 2007 fraud was up 26% on the previous year [83].

10.6.1.2 Dynamic Data Authentication

DDA is a more complex EMV protocol, used in Germany. It differs from SDA in that the cards are capable of doing public-key cryptography: each has an RSA public-private keypair, with the public part embedded in the card certificate. The cryptography is used for two functions. First, when the card is first inserted into the terminal, it's sent a nonce, which it signs. This assures the terminal that the card is present (somewhere). The terminal then sends the transaction data plus the PIN encrypted using the card's public key, and the card returns the application data cryptogram as before.

This provides a small amount of extra protection (though at the cost of a more expensive card — perhaps $1 each in volume rather than 50c). In particular, the PIN doesn't travel in the clear between the PIN pad and the terminal, so the Hungarian skimmer won't work. (The Tamil Tiger attack still works as in that case the shop assistant collected the PIN using the mark 1 eyeball.)

There are still significant vulnerabilities though. Even assuming that the cryptography is sound, that the software's properly written, that the interfaces are well-designed, and that the cards are too expensive to clone, the lack of any hard link between the public-key operation of proving freshness and accepting the PIN, and the shared-key operation of computing the MAC, means that the two-card attack could still be perpetrated.

10.6.1.3 Combined Data Authentication

CDA is the Rolls-Royce of EMV protocols. it's like DDA except that the card also computes a signature on the MAC. This ties the transaction data to the public key and to the fact that a PIN verification was performed (assuming, that is, the bank selected the option of including a PIN-verification flag in in the transaction data).

But the protocol still isn't bulletproof, as the customer has no trustworthy user interface. A wicked merchant could mount a false front over a payment terminal so that the customer would think she was paying $5 for a box of chocolates when in reality she was authorising a payment of $2500. (With over 200 approved terminal types, it's unreasonable to expect customers to tell a genuine terminal from a bogus one.) A bad merchant can also mount a *relay* attack. Two students of mine implemented this as a proof-of-concept for a TV program; a bogus terminal in a café was hooked up via WiFi and a laptop to a bogus card. When a sucker in the café went to pay £5 for his cake to a till operated by one student, his card was connected up to the false card carried by the other, who was lingering in a bookstore waiting to buy a book for £50. The £50 transaction went through successfully [401, 915].

An interesting possibility for relay attacks is to provide deniability in money-laundering. EMV transactions are now routinely used for high-value transactions, such as buying cars and yachts, and as they're between bank accounts directly they attract little attention from the authorities. So a bad man in London wanting to pay $100,000 to a crook in Moscow could simply arrange to buy him a BMW. With relaying, he could get an alibi by making this transaction just after a local one with witnesses; he might take his Member of Parliament out to a meal. If challenged he could claim that the car purchase was a fraud, and the police could have a hard time proving a transaction relay in the face of bank denials that such things happen.

There also are the usual 'social engineering' attacks; for example, a dishonest merchant observes the customer entering the PIN and then steals the card, whether by palming it and giving her back a previously-stolen card issued by the same bank, or by following her and stealing it from her bag (or snatching her bag). Such attacks have happened since the early days of bank cards. They can be automated: a bogus vending machine might retain a card and give back a previously-stolen one; or more pernicious still, use a card in its temporary possession to make a large online purchase. There is a nasty variant for systems that use the same card for online banking: the wicked parking meter goes online and sends all your money to Russia in the few seconds when you thought it was simply debiting you $2.50.

10.6.2 RFID

In the USA, where thanks to the Federal Reserve the incentives facing banks and merchants are less perverted, the banking industry has remained unconvinced that the multibillion-dollar costs of moving to EMV would be justified by any reductions in losses. Rather than moving to EMV, the industry has preferred to skip a generation and wait for the next payment method — so-called 'RFID' or 'contactless' payment cards.

Contactless payment has been a reality for a few years in a number of transport systems from London to Tokyo. When you buy a season ticket you get a 'contactless' smartcard — a device using short-range radio or magnetic signals to communicate with a terminal embedded in the turnstile. The automation allows a greater variety of deals to be sold than the traditional season ticket too; you can pay as you go and top up your credit as you need it. Turning this technology into a general-purpose payment instrument has a number of advantages.

One interesting new development is NFC — near-field communications. NFC is a means of building contactless/RFID communications capability into devices such as mobile phones. This means that your phone can double as your season ticket; at Japanese subway turnstiles, you can just touch your phone on the payment pad in order to get through. Small payments can be processed quickly and automatically, while for larger payments the phone can provide the trustworthy use interface whose lack is such a serious problem for EMV-style payment systems.

There are quite a few problems to be tackled. First, if RFID payment cards can also be used in traditional credit-card systems, then a bad man can harvest credit card numbers, security codes and expiry dates by doing RFID transactions with victims' cards as he brushes past them in the street — or by reading cards that have been sent to customers in the mail, without opening the envelopes [601].

Second, there are practical problems to do with RF propagation: if you have three cards in your wallet and you wave the wallet over a subway turnstile, which of them gets debited? (All of them?)

Third, our old friend the middleperson attack (and his evil twin the forwarding attack) return with a vengeance. When my students implemented the forwarding attack on EMV, they had to spend several weeks building custom electronics for the wicked reader and the bogus card. Once RFID and NFC become pervasive, making equipment is just a programming task, and not even a very tricky one. Any two NFC phones should be able to act in concert as the false terminal and the wicked card. And it appears that no-one's taking ownership of the problem of securing RFID payments; each of the players in the business appears to be hoping that someone else will solve the problem [56].

10.7 Home Banking and Money Laundering

After credit and debit cards, the third thread of electronic banking at the consumer-level is home banking. In 1985, the first such service in the world was offered by the Bank of Scotland, whose customers could use Prestel, a proprietary email system operated by British Telecom, to make payments.

When Steve Gold and Robert Schifreen hacked Prestel — as described in the chapter on passwords — it initially terrified the press and the bankers. They realised that the hackers could easily have captured and altered transactions. But once the dust settled and people thought through the detail, they realised there was little real risk. The system allowed only payments between your own accounts and to accounts you'd previously notified to the bank, such as you gas and electricity suppliers.

This pattern, of high-profile hacks — which caused great consternation but which on sober reflection turned out to be not really a big deal — has continued ever since.

To resume this brief history, the late 1980's and early 1990's saw the rapid growth of call centers, which — despite all the hoopla about the web — still probably remain in 2007 the largest delivery channel for business-to-consumer electronic commerce[3]. The driver was cost cutting: call centres are cheaper than bank branches. Round about 1999, banks rushed to build websites in order to cut costs still further, and in so doing they also cut corners. The bank-end controls, which limited who you could pay and how much, were abolished amid the general euphoria, and as we've seen, the phishermen arrived in earnest from 2004. Phishermen are not the only threat, although they appear to be the main one in the English-speaking world; in Continental Europe, there is some suspicion that keyloggers may be responsible for more account takeovers.

As I mentioned in the chapter on passwords, the main change is the increasing specialisation of gangs involved in financial crime. One firm writes the malware, another herds the botnet; another does the 'Madison Avenue' job of writing the spam; and there are specialists who will accept hot money and launder it. (Note that if it becomes too easy for bent programmers to make contact with capable money launderers, this could have a material effect on the fraud risk faced by systems such as SWIFT. It would undermine our current implicit separation-of-duty policy in that the techies who know how to hack the message queue don't understand how to get money out of the system.)

The hot topic in 2007 is how to stop phishermen getting away with money stolen from compromised bank accounts, and a phisherman faces essentially the same money-laundering problem as a bent bank programmer. Until May 2007, the preferred route was eGold, a company operated from Florida but with a legal domicile in the Caribbean, which offered unregulated electronic payment services. The attraction to the villains was that eGold payments were irreversible; their staff would stonewall bank investigators who were hot on the trail of stolen money. eGold duly got raided by the FBI and indicted.

[3]I'm not aware of any global figures, but, to get some indication, the UK has 6000 call centres employing half a million people; and Lloyds TSB, a large high-street bank, had 16 million accounts of whom most use telephone banking but under 2 million used online banking regularly in 2005.

The villains' second recourse was to send money through banks in Finland to their subsidiaries in the Baltic states and on to Russia; no doubt the Finnish regulators will have cleaned this up by the time this book appears. The third choice was wire-transfer firms like Western Union, and various electronic money services in Russia and the Middle East. I wrote a survey of this for the U.S. Federal Reserve; see [55].

At the time of writing, the favourite modus operandi of the folks who launder money for the phishermen is to recruit *mules* to act as cut-outs when sending money from compromised accounts to Western Union [545]. Mules are attracted by spam offering jobs in which they work from home and earn a commission. They're told they will be an agent for a foreign company; their work is to receive several payments a week, deduct their own commission, and then send the balance onward via Western Union. Money duly arrives in their account and they pay most of it onwards. After a few days, the bank from whose customer the money was stolen notices and reverses out the credit. The poor mule is left with a huge overdraft and finds that he can't get the money back from Western Union. In the English-speaking world, that's just about it; in Germany, mules are also prosecuted and jailed. (Even some German bankers consider this to be harsh, as the typical mule is an elderly working-class person who grew up under the communists in East Germany and doesn't even understand capitalism, let alone the Internet.) As the word gets round, mule recruitment appears to be getting more and more difficult — if we can judge from the rapidly increasing quantities of mule-recruitment spam during the middle of 2007. Note in passing that as the real victims of many phishing attacks are the poor mules, this implies that phishing losses as reported by the banks may be a significant underestimate.

Another thing we've learned from watching the phishermen over the past few years is that the most effective countermeasure isn't improving authentication, but sharpening up asset recovery. Of the £35m lost by UK banks in 2006, over £33m was lost by one bank. Its competitors assure me that the secret of their success is that they spot account takeovers quickly and follow them up aggressively; if money's sent to a mule's account, he may find his account frozen before he can get to Western Union. So the phishermen avoid them. This emphasises once more the importance of sound back-end controls. The authentication mechanisms alone can't do the job; you need to make the audit and intrusion-detection mechanisms work together with them.

Another thing we've learned is that liability dumping is not just pervasive but bad for security. The rush to online banking led many financial institutions to adopt terms and conditions under which their records of an electronic transaction are definitive; this conflicts with consumer law and traditional banking practice [201]. Unfortunately, the EU's 2007 Payment Services Directive allows all European banks to set dispute resolution procedures in their terms and conditions, and undermines the incentive to deal with the problems. The

ability of banks to blame their customers for fraud has also led to many sloppy practices. In the UK, when it turned out that people who'd accessed electronic services at Barclays Bank via a public terminal could be hacked by the next user pressing the 'back' button on the browser, they tried to blame customers for not clearing their web caches [1249]. (If opposing that in court, I'd have great fun finding out how many of Barclays' branch managers knew what a cache is, and the precise date on which the bank's directors had it brought to their attention that such knowledge is now essential to the proper conduct of retail banking business.)

10.8 Summary

Banking systems are interesting in a number of ways.

Bookkeeping applications give us a mature example of systems whose security is oriented towards authenticity and accountability rather than confidentiality. Their protection goal is to prevent and detect frauds being committed by dishonest insiders. The Clark-Wilson security policy provides a model of how they operate. It can be summarized as: *'all transactions must preserve an invariant of the system, namely that the books must balance (so a negative entry in one ledger must be balanced by a positive entry in another one); some transactions must be performed by two or more staff members; and records of transactions cannot be destroyed after they are committed'*. This was based on time-honoured bookkeeping procedures, and led the research community to consider systems other than variants of Bell-LaPadula.

But manual bookkeeping systems use more than just dual control. Although some systems do need transactions to be authorised in parallel by two or more staff, a separation of duty policy more often works in series, in that different people do different things to each transaction as it passes through the system. Designing bookkeeping systems to do this well is a hard and often neglected problem, that involves input from many disciplines. Another common requirement is non-repudiation — that principals should be able to generate, retain and use evidence about the relevant actions of other principals.

The other major banking application, remote payment, is increasingly critical to commerce of all kinds. In fact, wire transfers of money go back to the middle of the Victorian era. Because there is an obvious motive to attack these systems, and villains who steal large amounts and get caught are generally prosecuted, payment systems are a valuable source of information about what goes wrong. Their loss history teaches us the importance of minimizing the background error rate, preventing procedural attacks that defeat technical controls (such as thefts of ATM cards from the mail), and having adequate controls to deter and detect internal fraud.

Payment systems have also played a significant role in the development and application of cryptology. One innovation was the idea that cryptography could be used to confine the critical part of the application to a trusted computing base consisting of tamper-resistant processors — an approach since used in many other applications.

The recent adoption of various kinds of smartcard-based payment mechanism - EMV in Europe, RFID in the USA and Japan — is changing the fraud landscape. It opens up the possibility of more secure payment systems, but this is not at all guaranteed. In each case, the platform merely provides a toolkit, with which banks and merchants can implement good systems, or awful ones.

Finally, the recent history of attacks on electronic banking systems by means of account takeover — by phishermen, and to a lesser extent using keyloggers — presents a challenge that may over time become deeper and more pervasive than previous challenges. Up till now, banking folks — from the operations guys up to the regulators and down to the system designers — saw the mission as maintaining the integrity of the financial system. We may have to come to terms with a world in which perhaps one customer account in ten thousand or so has been compromised at any given time. Instead, we will have to talk about the resilience of the financial system.

Research Problems

Designing internal controls is still pre-scientific; we could do with tools to help us do it in a more systematic, less error-prone way. Accountants, lawyers, financial market regulators and system engineers all seem to feel that this is someone else's responsibility. This is a striking opportunity to do multidisciplinary research that has the potential to be outstandingly useful.

At a more techie level, we don't even fully understand stateful access control systems, such as Clark-Wilson and Chinese Wall. To what extent does one do more than the other on the separation-of-duty front? How should dual control systems be designed anyway? How much of the authorization logic can we abstract out of application code into middleware? Can we separate policy and implementation to make enterprise-wide policies easier to administer?

As for robustness of cryptographic systems, the usability of security mechanisms, and assurance generally, these are huge topics and still only partially mapped. Robustness and assurance are partially understood, but usability is still a very grey area. There are many more mathematicians active in security research than applied psychologists, and it shows.

Finally, if account takeover is going to become pervasive, and a typical bank has 0.01% of its customer accounts under the control of the Russian mafia at any one time, what are the implications? I said that instead of talking about the integrity of the financial system, we have to talk about its resilience. But what

does this mean? No-one's quite sure what resilience implies in this context. Recent experience suggests that extending the principles of internal control and combining them with aggressive fraud detection and asset recovery could be a good place for engineers to start. But what are the broader implications? Personally I suspect that our regulatory approach to money laundering needs a thorough overhaul. The measures introduced in a panic after the passage of the U.S. Patriot Act have been counterproductive, and perhaps emotions have subsided now to the point that governments can be more rational. But what should we do? Should Western Union be closed down by the Feds, as eGold was? That's probably excessive — but I believe that laundering controls should be less obsessive about identity, and more concerned about asset recovery.

Further Reading

I don't know of any comprehensive book on banking computer systems, although there are many papers on specific payment systems available from the Bank for International Settlements [114]. When it comes to developing robust management controls and business processes that limit the amount of damage that any one staff member could do, there is a striking lack of hard material (especially given the demands of Gramm-Leach-Bliley and Sarbanes-Oxley). There was one academic conference in 1997 [657], and the best book I know of on the design and evolution of internal controls, by Steven Root, predates the Enron saga [1082]. As the interpretation put on these new laws by the big four accountancy firms makes the weather on internal controls, and as their gold-plating costs the economy so much, it is certainly in the public interest for more to be published and discussed. I'll revisit this topic in Part III.

For the specifics of financial transaction processing systems, the cited articles [33, 34] provide a basic introduction. More comprehensive, if somewhat dated, is [354] while [525] describes the CIRRUS network as of the mid-80s. The transcript of Paul Stubbs' trial gives a snapshot of the internal controls in the better electronic banking systems in 2004 [975]; for conspicuous internal control failure, see for example [1038]. The most informative public domain source on the technology — though somewhat heavy going — may be the huge online manuals for standards such as EMV [429] and the equipment that supports it, such as the IBM 4758 and CCA [641].