**Chapter 12**

**Banking and Bookkeeping**

**Against stupidity, the Gods themselves contend in vain.**

– JC Friedrich von Schiller

**As a dog returneth to his vomit, so a fool returneth to his folly.**

– Proverbs 26:11

**12.1** **Introduction**

The cashless payment industry is one of the winners from the coronavirus pan-  
 demic, as people worldwide abandon cash in favour of card and phone payments.  
 The underlying banking systems range from payment card processing and home  
 banking through high-value interbank money transfers to the back-end book-  
 keeping systems that keep track of it all and settle up afterwards. There are  
 specialised networks for everything from stock trading to trade payments, many  
 of which are open to other companies too. Larger companies have internal book-  
 keeping and cash management systems that mirror many of the functions of a  
 bank.

Such systems matter to the security engineer for a number of reasons. First,

they’re a core professional competence. You need to understand transaction  
 processing to tackle the wider problems of fraud, and this chapter will give you a  
 road map. You also need to understand internal controls based on bookkeeping,  
 as these not only give early warnings when things go wrong, but also drive  
 corporate risk management. You have to be able to carry a conversation about  
 Gramm-Leach-Bliley, Sarbanes-Oxley and PCI DSS to have credibility with  
 your CFO. When you propose protection mechanisms, one of the ﬁrst things  
 you’re likely to be asked is how they’ll help executives discharge their ﬁduciary  
 responsibilities to shareholders.

Second, bookkeeping drove the computer industry. The ﬁrst computer out-

side the military and academia was the Leo, which did bookkeeping for the  
 Lyons chain of coffee houses from 1951. Banking rapidly became the most in-  
 tensive application area for computing, which spread into other ﬁrms via the

automation of bookkeeping from the 1960s. So the protection of bookkeeping  
 systems is of both historical and 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 paramount.  
 A banking system should prevent customers from cheating each other, or the  
 bank; it should prevent bank staff from cheating the bank, or its customers; and  
 the evidence it provides should be good enough that none of them can get away  
 with falsely accusing others of cheating. Banking and bookkeeping pioneered  
 the use of dual control, also known nowadays as multi-party authorisation.

Third, transaction processing systems – whether for $50 ATM withdrawals,

or $100m wire transfers – were the application that launched commercial cryptol-  
 ogy as a separate discipline outside the military. They drove the development of  
 encryption algorithms and protocols, as well as the supporting technology such  
 as smartcards. Many instructive mistakes were ﬁrst made (or at least publicly  
 documented) in the area of ﬁnancial cryptography.

Finally, many of the global-scale systems we’ve built this century were de-

signed to circumvent the checks and balances that had evolved over centuries  
 in the local and manual systems they replaced. Google’s mission was to make  
 all the world’s information available by disrupting the previous implicit and  
 explicit controls of locality, scale, conﬁdence and copyright. Uber planned to  
 become the global taxi company by circumventing taxi regulations in thousands  
 of towns and cities worldwide. It’s hardly surprising that a successful startup  
 often has to reinvent controls, whether under pressure from fraud and abuse, or  
 under pressure from lawmakers.

In this chapter, I’ll ﬁrst describe the bookkeeping systems used to track assets

and manage the risk of corrupt staff; such accounting systems are also used by  
 other companies of any size. I’ll then describe the international funds-transfer  
 systems used for interbank payments. Next, I’ll describe ATM systems, the

public face of banking, whose technology has also been adopted in applications  
 such as utility meters. I’ll follow with the story of credit cards, which have  
 become the main payment mechanism online. I’ll then move on to more recent  
 technical advances, including contactless payments, phone payments and open  
 banking.

**12.2** **Bookkeeping Systems**

Bookkeeping appears to have been invented in the Middle East in about 8500  
 BC, just after agriculture [1663]. 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 put what 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, sealed by rolling it with the pattern  
 of the warehouse keeper and then baked it in a kiln. 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 [1515]; 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 [1551]. By the Athens of  
 Pericles, a number of wealthy individuals were in business as bankers [772].

Figure 12.1: – 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 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 recognisably modern banks date to  
 this period; by having branches in a number of cities, they could ﬁnance trade.  
 But for ﬁrms to grow beyond the ability of the owner’s family to supervise  
 them directly, they had to hire managers from outside. The mechanism that  
 evolved to control the risk of fraud was *double-entry bookkeeping*. Historians  
 have found double-entry records created by Jewish merchants in twelfth-century  
 Cairo [1691], though the ﬁrst book on the subject did not appear until 1494 [522].

**12.2.1** **Double-entry bookkeeping**

The idea behind double-entry bookkeeping is simple: 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 to 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. In all but the smallest ﬁrms, the books

|  |  |  |
| --- | --- | --- |
|  |  |  |

were kept by different clerks.

We arrange things so that each branch can be balanced separately. Each

cashier will balance their cash tray before locking it in the vault overnight;  
 the debits in the cash ledger should exactly balance the physical banknotes  
 they’ve collected. So most frauds need the collusion of two or more people,

and this principle of *split responsibility*, also known as *dual control* or *multi-*  
 *party authorisation* (MPA), is complemented by audit. Not only are the books  
 audited at year end, but there are random audits too; inspectors may descend  
 on a branch at no notice and insist that all the books are balanced before the  
 staff go home.

Technology arrived in 1879, when the ‘Incorruptible Cashier’ patent of James

Ritty of Dayton, Ohio, introduced the cash register with a bell and a paper  
 tape. Ritty was a saloon owner whose employees stole money from him. He  
 sold his patent to John H. Patterson, who founded the National Cash Register  
 Company, which not only became a leading supplier of banking and bookkeeping  
 equipment, but spun off IBM, which dominated the computer industry until  
 Microsoft displaced it in the 1990s.

**12.2.2** **Bookkeeping in banks**

Banks were early adopters of computers for bookkeeping. Starting in the late  
 1950s and early 1960s with applications such as cheque processing, they found  
 that even the slow and expensive computers of the time were much cheaper than  
 armies of clerks. The 1960s saw banks offering automated payroll services to  
 their corporate customers. ATMs arrived en masse in the 1970s, with the ﬁrst  
 online banking systems in the 1980s; web-based banking followed in the 1990s.  
 Yet today’s slick online systems still rely on legacy back-office automation.

The law in the US, Europe and most developed countries requires not just

banks but all public companies to have effective internal controls, and makes  
 executives responsible for them. Such laws are the main drivers of investment  
 in information security mechanisms. Computer systems used for bookkeeping  
 typically claim to implement variations on the double-entry theme, but the  
 quality is variable. The separation-of-duty features may be just a skin in the  
 user interface, while the underlying data are open to manipulation by technical  
 staff. For example, if the ledgers are all just views of one single database,

then someone with physical access and a database editing tool might bypass  
 the controls. Staff may also notice loopholes and exploit them. For example,  
 one bank didn’t audit address changes, until a cashier found he could change a  
 customer’s address, issue an extra bank card, and change it back again [54]. So  
 we need to look at the mechanics, and banking is the natural place to start.

A traditional core banking system has a number of data structures: 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* of transactions that have been received from cash machines,  
 teller stations, merchant terminals and so on, but not yet posted to the ledgers;  
 and an *audit trail* that records who did what and when. The systems used by

|  |  |  |
| --- | --- | --- |
|  |  |  |

the large UK banks are relatively unchanged since the last century, though a  
 number of peripherals have been added, notably phone banking1.

The core banking software will apply the transactions from the journals to

the various ledgers and the account master ﬁle. So when a customer walks

into a branch and pays $100 into their savings account, the teller will make a  
 transaction that records a credit to the customer’s savings account of $100 while  
 debiting the same amount to the cash ledger recording the amount of money in  
 the drawer.

This was traditionally done overnight in a batch process but increasingly

involves real-time online processing, so things can go wrong more quickly. 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, some processing will stop, and inspectors will start looking for the  
 cause. So a programmer who wants to add to their own account balance has  
 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 a traditional business  
 had different ledgers managed by different clerks, so a banking data processing  
 shop will have different development teams 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. (The principle that a different team  
 runs production systems than the developers who wrote it is now coming under  
 strain in the new world of DevOps.)

**12.2.3** **The Clark-Wilson Security Policy Model**

Although such systems had evolved 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) [436]. In this  
 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* (IVPs) 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 that preserve balance. In  
 the general case, they maintain the integrity of CDIs. They also write enough  
 information to an append-only CDI (the audit trail) for transactions to be re-  
 constructed. Access control is by means of triples *(subject, TP, CDI)*, which  
 are so structured that a multi-party authorisation policy is enforced. In the  
 formulation in [47]:

1. the system will have an IVP for validating the integrity of any CDI;

1Most retail banking transactions nowadays are balance enquiries from phones, which are

typically dealt with by a front end that gets regular updates from the core system. This

minimises load on the core system, and also minimises the complaints when it goes down.

|  |  |  |
| --- | --- | --- |
|  |  |  |

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. First, unlike Bell-LaPadula, the Clark-

Wilson model involves maintaining state. In addition to 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 and  
 waiting for sign-off by a second.

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. This model doesn’t stop you paying cash into the  
 wrong bank account.

Third, the hard question remains, 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 difficult  
 to ﬁnd any systematic discussion in the accounting literature of how you design  
 internal controls.

What happens in practice is that the big four accountancy ﬁrms have a list

of controls that they push to their audit clients – a typical company may have  
 a checklist of about 300 internal controls that it has to maintain, depending on  
 what sector it’s in. These lists get steadily longer in response to incidents, fears,  
 and regulatory requirements. Many controls are formal compliance rather than  
 real risk reduction, and some are actually harmful. I discussed in section 3.4.4.3  
 how the big four auditors seized on NIST advice in the 1990s to get people to  
 change their passwords every month; at the time of writing (2020) they are still  
 pushing their audit clients to do this. Yet NIST retracted its advice years ago  
 in the face of the evidence, and Britain’s GCHQ also advises companies against  
 password aging.

A principled approach to internal control is possible, and indeed desirable.

In the following section, I try to distill the experience gained from working at the  
 coalface in banking and consultancy, and more recently in university governance.

**12.2.4** **Designing internal controls**

Over the years, various standards for bookkeeping and internal control have been  
 promoted by the accountancy profession, by lawgivers and by banking regula-

|  |  |  |
| --- | --- | --- |
|  |  |  |

tors. In the US, there’s the *Committee of Sponsoring Organizations* (COSO), a  
 group of accounting and auditing bodies [461]. However, self-regulation failed  
 to stop the excesses of the dotcom era, and following the collapse of Enron  
 there was intervention from US lawmakers in the form of the *Sarbanes-Oxley*  
 *Act* (SOX) of 2002. SOX regulates all US public companies, making senior exec-  
 utives responsible for the accuracy and completeness of ﬁnancial reports, whose  
 truthfulness CEOs have to certify; protecting whistleblowers, who are the main  
 source of information on insider fraud; and making managers responsible for  
 maintaining “adequate internal control structure and procedures for ﬁnancial  
 reporting”. It also demands that auditors disclose any “material weaknesses”.  
 Most of the compliance costs of SOX are reckoned to come from internal con-  
 trols. Earlier, the *Gramm-Leach-Bliley Act* (GLBA) of 1999 had liberalised bank  
 regulation in many respects but obliged banks to have security mechanisms to  
 protect information from foreseeable threats in security and integrity. Along  
 with HIPAA in the medical sector, and PCI DSS that I’ll discuss later in sec-  
 tion 12.5.2, GLBA and SOX have driven much of the investment in information  
 security and internal control. These regulations have helped consolidate the Big  
 Four accountancy ﬁrms’ inﬂuence over corporate policy on internal control.

In this section, our focus is on the technical aspects. Modern risk-management

systems typically require a company to identify and assess its risks, and then  
 build controls to mitigate them. The company will typically have a risk register  
 containing many pages of major risk items such as ‘loss of working capital due to  
 large unauthorised bank transaction by insider’ (I’ll discuss this in more detail  
 in section 27.2). Some of them will be mitigated using non-technical measures  
 such as insurance, but all should have a risk owner among the senior executives,  
 and a number of these risks will end up in the CIO’s lap2

The auditors’ work will be driven by the International Auditing and As-

surance Standard Board’s “International Standard on Auditing 315” [950]. ISA  
 315 focuses on the risk of a material misstatement in an organisation’s accounts,  
 whether due to error or to fraud. The auditors are supposed to understand the  
 business and its system of internal control; they will identify signiﬁcant accounts  
 (such as Cash), signiﬁcant assertions for each account (such as Existence) and  
 the signiﬁcant business processes (such as Sales) that impact them, along with  
 the controls those processes contain. They then work through the risk that

each assertion might be false and whether the risk is material. So how do you  
 engineer proper controls? The latest version of ISA 315 has quite a few pages  
 on this, but they are mostly somewhat general3, so their interpretation is often  
 down to the accountancy ﬁrms.

As we’ll discuss in Part 3, there are two basic approaches to assuring safety

against errors and security against attacks. You can work top-down, starting off  
 from the list of bad things you want to not happen, such as ‘large unauthorised  
 wire transfer’, then enumerating the possible causes and identifying controls to  
 mitigate the risks; or you can work bottom-up, starting off from things that  
 might fail, such as ‘a member of staff being blackmailed’, work out what harm  
 might result, and again identify appropriate controls. You may often have to

2For a description of risk governance in a UK bank, see the Financial Conduct Authority’s

report into the 2016 fraud against Tesco Bank [687], which I discuss in section 12.6.3.

3See paragraphs A6, A123–181, A198, A224–229 and Appendix 3 paragraphs 15–24

|  |  |  |
| --- | --- | --- |
|  |  |  |

use both approaches. When supporting audit, you need to pay attention to the  
 risks to assertions on which the ﬁnancial statements rely. However, you cannot  
 ignore other risks that might affect the ﬁrm’s ability to operate, such as the loss  
 of a data centre. The internal controls will not be all of your security posture.

Having identiﬁed those risks that need to be mitigated by separation of

duty, you can do this in two ways: *dual control*, also known as *multi-party*  
 *authorisation*, and *functional separation*.

In dual control, two or more principals act together to authorize a transac-

tion. 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 either to reach both locks (I’ll discuss this in detail in section 15.4).  
 The classic civilian example is when a bank issues a letter of guarantee, which  
 may undertake to carry the loss should a loan made by another bank go sour.  
 Guarantees are particularly prone to fraud. If you can get bank A to guar-  
 antee a loan to your business from bank B, then bank B is supervising your  
 account while bank A’s money is at risk. A crook with a forged or corruptly-  
 obtained guarantee can take their time to plunder the loan account at bank B,  
 with the alarm only being raised when they default and bank B asks bank A  
 for the money. You don’t want a single manager to be able to issue such an  
 instrument4.

With functional separation of duty, two or more staff members act on a trans-

action in complementary ways. The classic example is corporate purchasing. A  
 line 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 cheque; and the accounts manager signs  
 the cheque.

However, it doesn’t stop there. The line manager now gets a debit on their

monthly statement for that internal account, their 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.

The model can be summarised as *prevent – detect – recover*. The reliance

placed on each of these three legs will depend on the application. Where detec-  
 tion may be delayed, and recovery may therefore be difficult – as with corrupt  
 bank guarantees – you put extra effort into prevention, perhaps using dual con-  
 trol. Where it’s prevention that’s hard, you can make detection fast enough,  
 and recovery vigorous enough, to provide a deterrent. The classic example here  
 is that bank cashiers can easily take cash, so you count the money every day  
 before they go home.

Management control based on bookkeeping is not only one of the earliest

security systems; it has given rise to a lot of management science and civil law.

4Nowadays the issue is not just whether two managers might collude, or one of them

impersonate the other, but whether malware might take over both their accounts. I’ll discuss  
 this further in section 12.3.3.

|  |  |  |
| --- | --- | --- |
|  |  |  |

Controls work best where the roles are complementary parts of the existing  
 business process, and some processes have evolved over centuries to support  
 them. Controls are not only entwined with these processes, but exist in the  
 ﬁrm’s cultural context. In Swiss banks, there are two managers’ signatures on  
 almost everything, while Americans are much more relaxed. In most countries’  
 banks, staff can be moved randomly from one task to another, and are forced to  
 take a one-week or even two-week holiday, with no computer or building access,  
 at least once a year. This would not be acceptable in a university – but in  
 academia there’s a lot less to steal.

Designing an internal control system is highly interdisciplinary. 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 often neglected, and systems end up vulnerable  
 when helpful subordinates or authoritarian managers circumvent the control to  
 get their work done. It’s important to match the controls to the culture, and  
 motivate people to use them; the better run banks sell management controls  
 to staff as a means of protecting them against blackmail and kidnapping. As  
 we noted in Chapter 3, staff in an organisation only have so much compliance  
 budget – they’re only prepared to spend so much time and effort performing  
 security rituals that get in the way. Controls that become rituals may also

be practised for many years after their purpose has been forgotten or become  
 irrelevant. You have to understand all this and spend the compliance budget  
 wisely on achieving culturally feasible effects. A culture of limited trust of

close colleagues is particularly difficult to sustain (another reason why functional  
 controls split across business units may be more effective).

And just as you will try to require more than one banker to approve a large

transaction, you may want to require more than one engineer to approve code  
 to run on a live system. But this is hard to do thoroughly for a number of  
 reasons. First, many interfaces provide single points of failure. Second, split-  
 responsibility systems administration is just too tedious. With care you can  
 make it auditable5. Third, dual controls often require persistent state, which is  
 in tension with programmers’ wish to keep things simple by making transactions  
 atomic. And as that state needs to be managed, there are always some trusted  
 sysadmins who need full access in order to do their jobs. Fourth, as ﬁrms move  
 to integrating development and operations as DevOps, and then add security  
 to make it DevSecOps, they may end up with more trusted staff. At the very  
 least, the location of trust may change, as more of it shifts to the source code  
 review phase. Fifth, there are emergencies. The ATM system goes down at the  
 weekend, and the ATM team’s on-call engineer gets access to the live system  
 from home to ﬁx the bug. You log such accesses and get your auditors to stare at  
 the logs, as with the sysadmins. Finally, it’s inevitable that your top engineers  
 will be so much more knowledgeable than your auditors that they could do bad  
 things if they really wanted to.

So there are always engineers who could commit fraud. A sysadmin might

5Old-time banking systems were built on the IBM operating system MVS, which would

let the sysadmin do anything, except ﬁnding out which of their activities the auditor was  
 monitoring [224].

|  |  |  |
| --- | --- | --- |
|  |  |  |

create two shadow users who between them authorise a large payment, or a  
 payment system maintainer might pop an extra payment into the queue. Where  
 they get caught is when the balancing controls set off the alarm after a day or  
 two, and the money-laundering controls at the bank to which they wire the  
 money stop them getting away with very much. I’ll discuss this further in

section 12.3.3. The take-home is that functional controls along the *prevent –*  
 *detect – recover* model are often more important than shared control, as they  
 separate know-how as well as access. But for functional separation to work,  
 the mechanisms need to be engineered into the application, so they may be  
 proprietary, obscure and less well tested than the mechanisms that come with  
 operating systems. And there are limits to how much you can separate know-  
 how. Some people have to understand it all, such as the security architect and  
 the chief auditor.

The same analysis hold for the business processes themselves. Some people

end up having to take high-value decisions quickly and have to understand all  
 the aspects of a deal. At a real bank, you might ﬁnd thirty or forty people you  
 just have to trust – the CEO, the chief dealer, the top sysadmins and a number  
 of others. It’s important to know who they are, to minimise their numbers, to  
 pay them well, and to watch them discreetly.

A ﬁnal remark on dual control is that it gets fragile at organizational in-

terfaces. One example is that banks in California suddenly started ignoring

requests that cheques have two signatures after they installed new processing  
 equipment [1621]. Some organisations are unwilling to show competitors who’s  
 trusted to sign and for how much. And then there’s dispute resolution: ‘My two  
 managers say the money was sent!’ ‘But my two say it wasn’t!’

**12.2.5** **What goes wrong**

Theft and fraud can take many forms. Most thefts from the average company  
 are due to insiders, and automation seems to be making the incidents both rarer  
 and larger.

**12.2.5.1** **Insider frauds**

Back when most bankers worked in branches, banks in the English-speaking  
 world sacked some 1% of staff each year. The typical offence was minor embez-  
 zlement with a loss of a few thousand dollars. No-one found an effective way of  
 predicting which staff would go bad; previously loyal staff can be thrown off the  
 rails by shocks such as divorce, or be given a new manager they just can’t stand.  
 Losing a few hundred tellers a year was just a cost of doing business. These  
 numbers are falling now that most staff work in call centres; the customers they  
 deal with are allocated randomly to them, so it’s hard to collude with a friend.  
 It’s also harder nowadays for staff to sell customers’ personal information, since  
 staff have to walk a customer through security questions to get access to their  
 record. Staff at well-run banks are typically forbidden from taking phones or  
 even pens and paper into call centres so they can’t leak data to outsiders at any

|  |  |  |
| --- | --- | --- |
|  |  |  |

scale6.

Notable insider cases include:

*•* The biggest recent UK bank fraud was pulled off by a gangster from the  
 in 2016 for stealing at least £113m from business customers of Lloyds’  
 Bank in the UK during 2013–15, of which only £47m was recovered7. He  
 subverted two members of staff who spotted target companies – typically  
 medium-sized ﬁrms with over £1m in their accounts. Fizzy would then  
 phone up the business owner or ﬁnancial controller, claim to be from the  
 bank, ‘authenticate’ himself by reading them a couple of recent transac-  
 tions, and ask them to ‘authenticate’ themselves in return by computing  
 an authorisation code on their second-factor device. Before he did this,  
 he’d log on as them and set up a batch of payments for large ﬁve-ﬁgure  
 sums. The code he got from the victim would release the batch [820].

*•* A password reset clerk at HSBC conspired with persons unknown to  
 HSBC. The new password was used to transfer over $20 million to offshore  
 companies, from which it was not recovered. The clerk was a vulnerable  
 young man who had been employed on password reset after failing inter-  
 nal exams; the court took mercy, and he got away with ﬁve years [1569].  
 It was alleged that an AT&T employee had conspired to cover up the  
 transactions, but that gentleman was acquitted.

*•* One rapidly-growing bank fraud in the 2010s has involved spear-phishing  
 accounts. Owning two clerks’ PCs is simpler than suborning two clerks,  
 and if a ﬁrm’s PCs all have the same conﬁguration and update status,  
 it may not be too hard. As a bank may pay extra attention to large

transactions, the game is often to make a lot of four-ﬁgure payments before  
 the company notices. In the US, companies that don’t notice a fraudulent  
 payment the following day usually have no redress. A typical attack might  
 net half a million.

**12.2.6** **Executive frauds**

All the famous large ﬁnancial frauds – nine ﬁgures and up – have involved senior  
 insiders. The collapse of Barings Bank is a good example: managers failed to  
 control rogue trader Nick Leeson, blinded by greed for the bonuses his appar-  
 ent trading proﬁts earned them. Other examples include 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 Robert Maxwell’s looting of the Daily Mirror newspaper pension  
 funds in Britain. Either the victim’s executives were grossly negligent, as in the

6Such opsec rules are making it harder for call centres to get staff to work from home

during the Covid pandemic.

7Full disclosure: I acted as expert witness for one of the victim companies, and we had to

threaten to sue Lloyds to get our money back.

|  |  |  |
| --- | --- | --- |
|  |  |  |

case of Barings, or were the perpetrators, as with Equity Funding and Maxwell.  
 And these patterns repeat; for example, Wells Fargo was ﬁned $3bn in 2020 for  
 opening millions of accounts without the customers’ knowledge, just as in the  
 Equity Funding case [699].

Economists and accountancy professors analyse such issues as problems of

*agency*: a principal A hires an agent B to manage an asset and wants to know  
 how can B’s performance be monitored and assessed. The same principles apply  
 whether the principal is the bank’s CEO and the agent is a manager contem-  
 plating a fraud; or whether the principal consists of the shareholders and the  
 agent is the CEO. In theory, the internal controls and the internal audit depart-  
 ment are the tool used by the CEO to keep track of more junior staff, while  
 the external auditors are the tool used by the shareholders to keep track of the  
 CEO and the senior executives.

That’s the theory. The practice was analysed by Alexander Dyck, Adair

Morse and Luigi Zingales in a survey of 230 cases of corporate fraud against  
 quoted US companies between 1996 and 2004 [596]. Before Sarbanes-Oxley,

only a minority of frauds were revealed by the people mandated to spot them:  
 14% by the auditors and 6% by the SEC. Most were detected by actors with  
 other incentives: 19% by employees, 16% by industry regulators, 14% by ﬁ-  
 nancial analysts and 14% by the media. Stock-exchange regulators, commercial  
 banks and insurance underwriters are notable for their complete absence. Af-  
 ter Sarbanes-Oxley the performance of mandated actors improved slightly but  
 still to just over half the total. Their analysis of incentives shows that actors  
 with the strongest incentive to blow the whistle, such as short sellers, were least  
 active, while the most active, employees, often had negative incentives in that  
 they got ﬁred. This suggests that the dominating factor is who actually knows  
 what’s going on. Second, rewards promote disclosure: in addition to the ef-  
 fects of Sarbanes-Oxley, many government actors (such as the taxman) reward  
 whistleblowers, with positive effects.

In theory, external auditors are appointed by the board’s audit committee,

which is chaired by an external director; but who appoints the external direc-  
 tors? In my experience, the external directors tend to be friendly with the CEO  
 and the auditors go out of their way to schmooze the CFO8. They offer cheap  
 audits to get their foot in the door, and make their real money from consul-  
 tancy; this was a structural problem for decades, and eventually in February  
 2020, the UK Financial Reporting Council ordered audit and consultancy to be  
 separated [1049]. The big audit ﬁrms have a pernicious effect on the information  
 security world by pushing their own list of favourite controls, regardless of the  
 client’s real risks. They maximise their income by nit-picking and compliance;  
 the Sarbanes-Oxley regulations cost the average US public company over $1m  
 a year in audit fees.

Quite apart from the pure economic incentives, bosses ﬁnd it hard to cope

with evidence that senior colleagues are incompetent or dishonest. There’s a  
 whole literature on information avoidance, which I mentioned in section 3.2.4:  
 people are reluctant to learn things that will cause them pain, stress or extra

8The legal inﬁghting following the collapse of Enron destroyed its auditors Arthur Ander-

sen, reducing the ‘big ﬁve’ audit ﬁrms to the ‘big four’; now auditors go out of their way to  
 avoid liability for fraud.

|  |  |  |
| --- | --- | --- |
|  |  |  |

work. And risks that managers are unwilling to confront, they are often unable  
 to control. No-one at Barings wanted to think that their star dealer Nick Leeson  
 might be a crook; and pop went the bank. Such risks are not being mitigated  
 by technology; if anything they may be growing.

**12.2.6.1** **The Post Office case**

Executives can also be unwilling to believe that anything might be going sys-  
 tematically wrong with their accounting systems. Even if they suspect, there’s  
 a social reﬂex to close ranks under criticism, and lawyers may advise clients to  
 just deny everything.

The case worth studying here is the failure of the Post Office accounting

system in the UK. The Post Office doesn’t just ship letters but is a signiﬁcant  
 ﬁnancial institution too, most of whose branches are run by sub-postmasters –  
 typically shopkeepers with a franchised Post Office counter on their premises. To  
 control them, the Post Office built an accounting system called Horizon, which  
 had multiple bugs that caused many franchisees to be charged money they didn’t  
 owe. Thousands of people had their lives ruined; some lost their businesses and  
 were bankrupted, some staff were wrongly ﬁred, and several people were jailed  
 for frauds they did not commit. Eventually 587 sub-postmasters sued the Post  
 Office, and in December 2019 they won an apology and £58m. The judge found  
 that Horizon ‘was not remotely robust’ [185].

This is the ﬁrst and only case, so far as I know, where an accounting system

has been subjected to a proper test in aggressive litigation. Many legal systems  
 presume that accounting systems are working properly unless someone can pro-  
 duce evidence to the contrary, and this can be hard: a lot of the legal effort went  
 into forcing the Post Office to give the claimants access to the software and its  
 documentation so it could be examined by their experts. Incidentally, the total  
 losses to franchisees appear to be in the mid-hundreds of millions; they’ll get  
 maybe £11m of the £58m settlement, with the rest going to the lawyers and to  
 the hedge fund that bankrolled the litigation. Most staff at the Post Office took  
 a pay cut while the CEO Paula Vennels, an ordained minister, got a substantial  
 raise [354]. She eventually left. It may be that the software supplier, Fujitsu,  
 will end up paying for the settlement, but that may require further litigation.

**12.2.6.2** **Other failures**

Most accounting system failures are less spectacular, but there are many failures  
 that have signiﬁcant effects on the ability of ﬁnancial and other ﬁrms to operate.  
 We’ll see more examples as we work through payments in this chapter and other  
 applications in later chapters, but here’s a start sample.

1. As computer systems get more complex over time, they accumulate cruft

that makes them more fragile and harder to maintain. Software engineers  
 refer to this as *technical debt*: it means that changes become slower and  
 more expensive, and recovery from failures can be complex [41]. Book-  
 keeping systems are no exception. For example, in June 2012, 6.5 million

|  |  |  |
| --- | --- | --- |
|  |  |  |

customers of the Natwest Bank had service disrupted for several weeks fol-  
 lowing a software upgrade that went wrong and had to be reversed. People  
 were stranded overseas with no money and some companies couldn’t make  
 payroll. The bank was ﬁned £42m [686]; it was then largely owned by the  
 UK government as it had gone bust in the crash of 2008. Had the service  
 failure gone on another week, it might well have gone bust again, costing  
 taxpayers tens of billions and causing widespread disruption. So the fear  
 of a catastrophic failure closing a money-centre bank is a real one. But  
 replacing a crufty old core banking system with a new one is a major  
 project taking years and costing nine ﬁgures, with its own strategic risks.  
 As a young man I worked on a couple of such projects: they have their  
 nail-biting moments.

2. We ﬁnd similar project risks further down the food chain. Our univer-

sity’s accounting system was replaced in the early 2000s, and a project  
 that should have cost £3m cost £11m instead. We ended up suing the  
 accountancy ﬁrm that installed it, and published a detailed report of what  
 went wrong [691].

3. The system is still, years later, a pain to use, and the reason why may be

of interest. At our university, 35 ﬁnance-office staff have more say in the  
 design of the ﬁnance system than 1,500 professors. The clerks care more,  
 as they use it all the time, while we professors might use it for an hour  
 or two a week. The time saved by clerks is less than the time wasted by  
 professors, but the concentrated interest usually wins.

So even if your bookkeeping system uses a standard core that enforces the

basic Clark-Wilson properties of balance and integrity, there’s still a lot to go  
 wrong.

**12.2.6.3** **Ecological validity**

And it’s not enough to just check that the books are internally consistent. You  
 also need to check that they correspond to external reality. The series of scandals  
 that shaped modern audit requirements and practice began with the collapse  
 in 1938 of McKesson and Robbins, a well-known drug and chemical company  
 with reported assets of $100m 9. It turned out that 20% of the recorded assets  
 and inventory did not exist. The president, Philip Musica, turned out to be a  
 bootlegger with 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 had accepted the McKesson

account without making enquiries about the company’s bosses; they failed to  
 check inventories, verify accounts receivable with customers, or think about  
 separation of duties within the company [1616].

The famous case for the next generation was the salad oil scandal of 1963,

involving the bankruptcy of the Allied Crude Oil Reﬁning Corporation and the  
 prosecution by Robert F. Kennedy of its CEO, Tino de Angelis. Allied had bor-  
 rowed millions from American Express and others against tanks of soybean oil

9About $1.8bn in 2020 dollars

|  |  |  |
| --- | --- | --- |
|  |  |  |

that were actually mostly water, and used this to trade heavily in futures [1442].  
 American Express stock dropped by 50% after a whistleblower told it of the  
 fraud; it lost $58m. (Warren Buffett then bought 5% of the company and made  
 a fortune.)

The requirement that all big ﬁrms be audited has entangled audit ﬁrms in

pretty well every major ﬁnancial scandal. I already mentioned Enron, whose  
 failure in 2001 led to the Sarbanes-Oxley Act, and then there was the ﬁnancial  
 crisis in 2008 caused in part by trading complicated ﬁnancial derivatives that  
 turned out to be based on near-worthless mortgages. And one issue with the  
 blockchain systems currently being promoted for some payment and bookkeep-  
 ing applications is that while the mathematical structure may give guarantees  
 of consistency and consensus, there is no information whatsoever about whether  
 the assets referred to are sound, or even exist. So you might be somewhat scep-  
 tical when you see a bank talking about a blockchain to register mortgages,  
 on which smart contracts will allow ﬁnancial innovation. I’ll return to this in  
 section 20.7.

The most recent scandal as this book went to press in July 2020 was Wire-

card. A payment service ﬁrm, it had started out processing card payments

to porn sites, online casinos and other merchants that normal banks wouldn’t  
 touch. It grew rapidly to displace Commerzbank in the Dax 30 – the index  
 of Germany’s 30 biggest quoted companies, and was celebrated in Germany as  
 a rare local ﬁrm able to challenge Silicon Valley. But in June 2020, as it was  
 attempting to buy Deutsche Bank (Germany’s largest bank, with a market cap  
 of about $20bn), Wirecard’s auditors EY disclosed that a quarter of its claimed  
 assets, some e2.1bn supposedly held in the Philippines, could not be found.  
 (EY had failed to verify its bank statements with its bankers for three years,  
 relying instead on ‘screenshots’ provided by the company itself [1834].) The ﬁrm  
 ﬁled for bankruptcy and its CEO, Markus Braun, was arrested. A string of ﬁn-  
 tech startups that used it to process payments stopped trading, leaving millions  
 of cardholders inside and outside Germany unable to access their money. Yet  
 investors and regulators had ignored numerous red ﬂags, going back as far as  
 2008 [1256]. Worse, when the Financial Times published an analysis in 2019 of  
 Wirecard’s dubious accounting practices – pointing out that its Dubai subsidiary  
 seemed to have no customers, that the address of one alleged Philippines sub-  
 sidiary was a small bus company, that another was the home of a retired seaman,  
 and that whistleblowers in its Singapore subsidiary had reported they were being  
 ordered to cook the books [1283] – the German regulator BaFin had responded  
 not by investigating the company but by starting a criminal investigation of the  
 journalists and banning short selling of the company’s shares [610]. BaFin had  
 for some years defended the company against critics rather than investigating  
 their criticisms. This was one of the largest frauds in European history, de-  
 stroying over e20bn in apparent shareholder value, as well as public conﬁdence  
 in German ﬁnancial regulation. En route Wirecard had taken in ﬁrms such as  
 Moodys, Credit Suisse and Softbank. It was quite astonishing to see how little  
 the lessons of McKesson and Robbins had been heeded; checking overseas cash  
 balances really should have been audit 101. Yet the audit industry has persis-  
 tent structural problems, ranging from the fact that auditors sell to CFOs to  
 the fact that almost all the work is done by juniors [703].

|  |  |  |
| --- | --- | --- |
|  |  |  |

**12.2.6.4** **Control tuning and corporate governance**

The main reason internal control structures tend to be conservative, expensive  
 and ineffective is that while in theory organizations develop them in the light of  
 experience, in practice this experience is relayed through the auditor cartel. In  
 theory there is some governance behind this. There’s a survey of internal audit  
 standards in [**?**]; the most inﬂuential is the Risk Management Framework from  
 the *Committee of Sponsoring Organizations* (COSO), a group of US accounting  
 and auditing bodies [461]. This is one yardstick by which your system will

be judged if it’s used in the US public sector or by companies quoted on US  
 equity markets. The COSO model is targeted not just on internal control but on  
 the reliability of ﬁnancial reporting and compliance with laws and regulations.  
 Its basic process is an evolutionary cycle: in a given environment, you assess  
 the risks, design controls, monitor their performance, and then go round the  
 loop again. COSO emphasizes soft aspects of corporate culture more than hard  
 system design issues so may be seen as a guide to managing and documenting  
 the process by which your system evolves. In theory, its core consists of senior  
 management checking that their control policies are being implemented and  
 achieving their objectives, and modifying them if not. In practice, the auditors  
 have captured it.

The Information Systems Audit and Control Association (ISACA), which

administers the Certiﬁed Information Systems Auditor (CISA) exam, has a  
 reﬁnement of COSO known as the *Control Objectives for Information and re-*  
 *lated Technology* (CobiT) which is more international [946]. It extends from the  
 technical aspects of internal audit to personnel management, change control and  
 project management. More concrete standards emerge from auditors’ interpre-  
 tation of speciﬁc sectoral regulations, such as Sarbanes-Oxley for US publicly-  
 listed companies, Gramm-Leach-Bliley for US ﬁnancial-sector ﬁrms, HIPAA for  
 US healthcare providers and GDPR for the personal information of residents  
 of EU member states. And, as we noted in the chapter on banking and book-  
 keeping, the standards set by the PCI trade association govern data relating  
 to payment cards. There’s also ISO 27001 on security management. Whatever  
 sectors you or your customers operate in, it’s worthwhile paying attention to  
 evolving cybersecurity standards. Many of these are standards because every-  
 one can agree on them, so they’re by no means sufficient. Pretty well every big  
 breach involves a ﬁrm with ISO 27001 certiﬁcation; the auditors said something  
 was OK when it wasn’t. We’ll return to this in section 28.2.9.

**12.2.7** **Finding the weak spots**

If you are ever responsible for security in an organisation, you should not just  
 think about which components might, by their failure, cause a bad enough loss  
 to make a material difference to the bottom line. You need to think about

the people too, and their external relationships. 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 your call-centre staff be selling data from the accounts  
 they’ve dealt with to a phishing gang who use this data to impersonate your  
 company to your customers? Lots of things can go wrong. You have to ﬁgure  
 out which of them matter, and how you get to ﬁnd out. Remember the old  
 experience of 1% of staff falling into temptation every year. Remember that a  
 trusted person is one who can damage you. Who can damage you, and how?  
 This is what a control maintainer must constantly think about.

The lessons to be learned include the following.

*•* Maintaining effective controls is hard in a changing environment and needs

*•* If you rely on complaints from customers or staff to alert you to fraud and  
 for you to listen to them. Many companies cut costs by being hard to  
 contact, but this has consequences.

*•* The main exposure is to the company’s own staff and contractors, so  
 to defraud the company, how would you do it?’

*•* Don’t just think in terms of transactions and processes, but about people,  
 Do you expect people to keep each other honest without any motivating  
 structure, and nothing but risk for whistleblowers?

*•* No security policy can achieve full compliance, as workarounds will be

*•* These workarounds naturally create vulnerabilities, so you’d better design

*•* You’d better have a working relationship with the ﬁrm’s executive leader-  
 to your responsibilities, and so they understand what you’re doing too.

There will always be residual risks. Managing these residual risks remains

one of the hardest and most neglected of jobs. It’s an extremely bad idea to  
 adopt a doctrine that some particular system is foolproof – because if you assign  
 its failure an a priori probability of zero, then evidence won’t shift it and things  
 could go badly wrong when it eventually fails. More generally, you need to help  
 the ﬁrm learn from experience. And experience means not just loss history:  
 controls that get in the way need to be identiﬁed and improved. If you’re seen  
 as contributing to proﬁts rather than just as another compliance burden, you’ll  
 be listened to a lot more. For example, if you can ﬁx the password reset function  
 so it needs fewer staff, or improve the fraud engine so that the company’s website  
 rejects fewer shopping baskets, the board will listen to you a lot more readily.

Finally, your risk management systems will have to pay some homage to one

or more compliance regimes, depending on the industry. The international stan-  
 dard ISO 27001 on security management is used in some industries: it demands  
 that you analyse the risks systematically and subject the unacceptable ones to

|  |  |  |
| --- | --- | --- |
|  |  |  |

some form of risk treatment (control, avoidance, transfer); and have a man-  
 agement process to ensure that the controls are updated. In many companies,  
 this will be driven by your auditors anyway. And there are many sector-speciﬁc  
 regulatory regimes to deal with. In healthcare you have to worry about HIPAA  
 (see section 10.4); and as for banking and payments, we turn to that next.

**12.3** **Interbank Payment Systems**

When people think of electronic fraud, they often envisage a Hollywood scene in  
 which crafty Russian hackers break a bank’s codes and send zillion-dollar wire  
 transfers to tax havens. Systems for transferring money are indeed a crime tar-  
 get, and have been for a century and a half. We’ll look ﬁrst at the systems used  
 to transfer money between banks, and then at those used by bank customers,  
 whether individuals or merchants.

**12.3.1** **A Telegraphic History of E-commerce**

Many people assume that e-commerce is something invented in the mid-1990s.  
 But it goes back much further.

Governments used visual signalling from classical times, including heliographs

(which used mirrors to ﬂash sunlight at the receiver), semaphores (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. After the Napoleonic War, the French government  
 opened its heliograph network to commercial use, and soon the ﬁrst frauds took  
 place. 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 mak-  
 ing errors in transmissions that they could observe from a safe distance. Other  
 techniques were devised to signal the results of horseraces. Bookies learned to  
 ‘call time’ by a clock, rather than waiting for a result and hoping that they were  
 the ﬁrst to hear it.

From the 1760s to the 1840s, the electric telegraph was developed by a num-

ber 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 in many ways like the Internet boom of the late 1990s.

Banks were the ﬁrst big users, and found that they needed 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 [1818]. 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 nineteenth-century answer was trusted intermediaries – principally  
 banks who helped business manage risk using references, guarantees and letters  
 of credit.

By the 1970s, bankers started to realise that this worthy old Victorian system

was due for an overhaul.

First, as I noted earlier in section 5.2.4, most test-key systems were vulner-

able to cryptanalysis; someone who observed a number of transactions could  
 gradually work out the key material.

Second, the test key system didn’t support dual control. The secret tables

were kept in a safe, and two clerks would sit together to work out a test and  
 check it; but there was nothing really to stop staff members working out tests  
 for unauthorised messages at the same time.

Third, the real concern was cost and errors. The use of manual cryptography

meant that each transaction was typed on a keyboard at least three times: once  
 into the paying bank’s computer which would print out a transaction in the  
 telex room, where a test was computed manually; then a second time to send a  
 telex to the receiving bank, who would check the test manually; then the third  
 time as that bank fed it into their own computer. Errors were much more of a  
 problem than frauds. Surely the payments could ﬂow directly from one bank’s  
 computer to another?

**12.3.2** **SWIFT**

A consortium of banks set up the Society for Worldwide Interbank Financial  
 Telecommunications (SWIFT) in the 1970s to provide a more secure, efficient  
 and controllable mechanism for sending payment instructions between member  
 banks. It can be thought of as an email system with built-in authentication and  
 non-repudiation services, plus optional encryption. It’s used to ship trillions  
 of dollars round the world daily, and its design has been copied in systems  
 processing the title to many other kinds of asset, such as the bills of lading that  
 prove ownership of ships’ cargoes.

The design constraints are interesting. The banks did not wish to trust

SWIFT to the point that its employees could forge bank transactions. The au-  
 thenticity 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 could not use  
 digital signatures, as they hadn’t been invented yet. Finally, the banks had to  
 be able to enforce auditable dual controls over interbank transactions.

The design of SWIFT I is summarized in Figure 12.2. Authenticity of mes-

sages was assured by computing a message authentication code (MAC) at the  
 sending bank and checking it at the receiving bank. The keys used to be man-  
 aged using *bilateral key exchange*: whenever a bank set up a relationship over-  
 seas, the senior manager who negotiated it would exchange keys with his op-  
 posite number, whether in a face-to-face meeting or afterwards by post to each  
 others’ home addresses. There were two key components to minimize the risk of

|  |  |  |
| --- | --- | --- |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| R  Key� Ba  Bra |  |  | Logs�  P�  nk� Key�  nch� |
| P� Swift� R  nk� Ba  nch� Bra | |

Figure 12.2: – architecture of SWIFT

compromise, with one sent in each direction (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).  
 Authentication was not enabled until both banks conﬁrmed that the other’s key  
 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 was supposed to be a trade secret,  
 but as banks like their security mechanisms to be international standards, people  
 ﬁgured out to look at ISO 8731 [1631]. Pretty quickly, an attack was found and  
 published in [1545]. Fortunately, this attack takes over 100,000 messages to

recover a key – which was too large for a practical attack on a closed system  
 and gave the banks time to migrate to more modern mechanisms.

Although SWIFT itself was not trusted for 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 its  
 country, which also logged them. The RGPs were generally run by different  
 service ﬁrms. Thus any banker wishing to dishonestly repudiate a transaction  
 would have to subvert not just the local SWIFT application and its surrounding  
 controls, but two independent contractors in different countries. And logs are  
 easier for judges to understand than cryptography.

Conﬁdentiality was an optional add-on. It was provided by line encryption

devices between the banks and the RGP node, and between these nodes and  
 the main SWIFT processing sites. Keys were hand-carried 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 mechanisms10.

10In one country, a bank that attempted to install line encryptors found noise appearing on

the line after a few hours. This only appeared on the live line, not the backup one, only after  
 a delay, and swapping the equipment between the two lines didn’t help. The bank realised  
 that the local spooks wouldn’t tolerate encryption and gave up.

|  |  |  |
| --- | --- | --- |
|  |  |  |

Dual control was provided either by specialized terminals or by software

packages that could be integrated with other bank systems. 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 it11. There’s a further functional control  
 in that you reconcile accounts by checking transactions against statements every  
 day. So a bogus payment instruction that gets past the entry controls should  
 result in an alarm the following business day.

**12.3.3** **What goes wrong**

SWIFT I ran for twenty years without a single report of external fraud against  
 the system itself. In the mid 1990s, after the attack on the MAC algorithm was  
 published, it was enhanced by adding public key mechanisms: SWIFT II still  
 used bilateral key exchange, but with MAC keys shared between correspondent  
 banks using public-key cryptography and the MACs themselves further pro-  
 tected by a digital signature. The key-management mechanisms were ensconced  
 as ISO 11166, and there was some debate over the security of this architec-  
 ture [112, 1631]. Quite apart from the centralization of trust brought about by  
 the adoption of public key cryptography – in that a central certiﬁcation author-  
 ity could falsely certify a key as belonging to a bank when it doesn’t – at least  
 one early deployment adopted 512-bit public keys because of U.S. export con-  
 trols, and by 2000 at least one RSA public key of this length had been factored  
 surreptitiously by a group of students [43]. Bilateral key exchange was replaced  
 in 2009 with a new system whose cryptographic mechanisms are proprietary.  
 The messaging standard is being replaced by ISO 20022.

A political row arose once the crypto started to be toughened up and to of-

fer conﬁdentiality by default. The New York Times disclosed in June 2006 that  
 the NSA was accessing the entire transaction stream, whereupon the NSA sim-  
 ply demanded access to everything. This caused a confrontation with privacy-  
 conscious Europeans, but eventually after President Obama succeeded President  
 Bush, the EU agreed a treaty under which the US Treasury Department can  
 serve subpoenas on Swift [341]. Payments within Europe were supposedly ex-  
 cluded, but since such payments were targeted, and Ed Snowden revealed the  
 scale of collection, the issue has been raised repeatedly by the European Parlia-  
 ment and by privacy authorities12.

Criminal (as opposed to governmental) attacks on interbank systems have

not involved the payment mechanisms themselves but the surrounding business  
 processes. It does happen from time to time that a bank programmer inserts a  
 bogus message into the processing queue, but it usually fails because he doesn’t  
 understand the business process. How an international wire transfer actually  
 works is that 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

11As the checker can modify the payee and the amount, this is really only dual control, not

triple control – and the programmers who maintain the interface can always attack the system  
 there, unless you can maintain separation of duty on the systems side too.

12One might ask why banks don’t just build new systems with end-to-end crypto, but bank

regulators demand access to all message traffic between banks, and some traffic within banks,  
 to enforce rules against insider trading.

|  |  |  |
| --- | --- | --- |
|  |  |  |

have both balances and credit limits, and as payments may have to go through  
 one or more correspondent banks, large payments need human interventions to  
 make the money available. There are also ﬁlters that look for large transactions  
 so that the bank can report them to the money-laundering authorities [75]. So  
 a naive programmer who sneaks in a bogus transaction to an account he’s set  
 up at a Swiss bank usually gets arrested when he turns up to collect the cash.

The most famous attack carried out via Swift was in 4–5 February 2016 when

North Korean agents stole $63m from the Bank of Bangladesh. They appear to  
 have used Dridex malware to steal the credentials of bank staff and then ordered  
 four transactions that transferred $81m from the bank’s account at the Federal  
 Reserve in New York to the Philippines, of which only $18m was recovered;  
 the rest got laundered through a local casino. A further 30 transactions for a  
 total of $851m were ﬂagged for manual review by the Fed and not sent; another  
 for $20m was sent to Sri Lanka which was recovered after the paying bank  
 noticed a spelling error and stopped payment. This was not actually an attack  
 on Swift, but an attack on the Bank of Bangladesh’s own gateway to the Swift  
 system [858].

But if your life’s goal is to get rich from bank fraud, you’re probably better

off getting a law degree and working as a bank manager rather than messing  
 about with computers. In fact, most signiﬁcant frauds have exploited procedural  
 vulnerabilities rather than technical attacks.

*•* Perhaps the ﬁrst famous wire fraud was in 1979 when Stanley Rifkin, a  
 ciﬁc National Bank. He got round the controls by agreeing to buy a large  
 shipment of diamonds from a Russian government agency in Switzerland.  
 He observed an authorization code used internally when dictating trans-  
 fers to the wire transfer department, and used it over the telephone –  
 a classic example of dual control breakdown at a system interface. He  
 gave himself extra time to escape by doing the deal just before a US bank  
 holiday. Where he went wrong was in not planning what to do after he  
 collected the stones. If he’d hidden 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 went on the run and got caught.

*•* A fraud of a slightly different type took place in 1986 between London  
 two exchange rates, and in one bank the manager responsible for decid-  
 ing 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 central bank sent the police round. When he saw  
 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 had no exchange controls, exchange  
 control fraud wasn’t an offence, so he couldn’t be extradited. This is per-  
 haps the only case I know where the perp not only got away with several  
 million but also got to brag about it.

|  |  |  |
| --- | --- | --- |
|  |  |  |

*•* I’ve seen bad guys getting away with fraud using a letter of guarantee.  
 guarantee a loan to a company in another. This can be set up as a SWIFT  
 message, or even a paper letter, between the two banks. 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 dis-  
 appearing. Only when the lending bank realises that the loan has gone  
 sour and tries to call in the guarantee is the forgery discovered. Then you  
 can end up with a computer forensics case as two banks argue over whose  
 fault it was.

The lesson is to be alert to anything that can defeat dual control. But you

need to see this in a broader context. It’s not just the technical problems of  
 systems administration, interfaces or even shared-control crypto: the core is  
 the business process design. And quite often, critical transactions don’t appear  
 as such at a casual inspection. Proper split control usually needs functional  
 separation, and for that you need to really understand the application in its  
 social and economic context.

**12.4** **Automatic Teller Machines**

Our second set of lessons emerges from studying payment cards. This story  
 has at least four components: ﬁrst, *automatic teller machines* (ATMs); second,  
 credit cards; third, the chip cards that have taken over as both debit and credit  
 cards since the mid-2000s; and fourth, contactless payments including phone  
 banking.

ATMs were one of the most inﬂuential technological innovations of the 20th

century. They were devised in 1938 by the inventor Luther Simjian, who also  
 thought up the teleprompter and the self-focusing camera. He persuaded Citi-  
 corp to install his ‘Bankamat’ machine in New York in 1939, but 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’ [1743]. Its comeback was in 1967, when a machine made by De La Rue  
 was installed by Barclays Bank in Enﬁeld, London. According to the World  
 Bank, there are now over 2.4m machines, or 41 per 100,000 adults [2041]. Card  
 payments with PINs are now used in many terminals in shops, and the tech-  
 nology, including block ciphers, tamper-resistant hardware and the supporting  
 protocols, ended up being adapted for 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.

**12.4.1** **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’s magnetic strip contains

|  |  |  |
| --- | --- | --- |
|  |  |  |

|  |  |
| --- | --- |
| PAN:  PIN key *KP*:  Result of DES *{PAN}KP* :  *{N}KP* decimalized:  O↵set:  Customer PIN: | 8807012345691715  FEFEFEFEFEFEFEFE  A2CE126C69AEC82D  0224126269042823  0224  6565  6789 |

Figure 12.3: – IBM method for generating bank card PINs

the customer’s *primary account number* (PAN) and an expiry date. A secret  
 key, called the ‘PIN key’, is used to encrypt the PAN, then decimalize it and  
 truncate it. The result of this operation is called the ‘natural PIN’; an offset  
 can be added to give the PIN which the customer must enter. The offset has no  
 cryptographic function; it just enables customers to choose their own PIN. An  
 example of the process is shown in ﬁgure 12.3.

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. So  
 each ATM could 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.  
 From the mid-1990s, networks became more dependable and ATMs have tended  
 to operate online only, which simpliﬁed the design. Starting in 2003, magnetic  
 strips were supplemented with smartcard chips, followed by contactless payment  
 from 2012; I’ll describe these enhancements in later sections. But the basic

principle remains: PINs are generated and protected using cryptography.

A cryptographic processor, known as a *hardware security module* (HSM), is

kept in the bank’s server room and manages customer PINs so as to enforce a  
 dual-control policy.

1. Operations on the clear values of customer PINs, and on the keys used

|  |
| --- |
| to protect them, are always done in a *secure cryptographic device* (SCD),  so that no member of the bank’s sta↵ ever gets to see a PIN other than  their own. SCDs include the HSMs in the bank server room13 along with  crypto modules in ATMs and other PIN-entry devices. |

2. Thus, for example, the cards are personalized in a facility with machines

to emboss the card, encode the mag strip and initialise the chip, while the  
 PIN mailers are printed in a separate facility containing a printer attached  
 to an HSM. They’re mailed out a few days apart.

3. A *terminal master key* is supplied to each ATM in the form of two printed

components, which are carried to the branch by separate people, input  
 at the ATM’s rear keyboard, and combined to form the key. Similar

ceremonies (but with three people) are used to set up master keys between  
 banks and network switches such as VISA.

13Or nowadays, also in a cloud service provider or other service contractor

|  |  |  |
| --- | --- | --- |
|  |  |  |

4. If ATMs perform PIN veriﬁcation locally, then the PIN key is encrypted

under the terminal master key and sent to the ATM. Keys are stored in a  
 local SCD – a tamper-resistant chip next to the keyboard – which either  
 veriﬁes PINs as they’re entered or encrypts them so they can be sent from  
 the ATM to a central HSM for checking.

5. If the bank’s ATMs are to accept other banks’ cards, then the PIN will

be encrypted in the ATM’s SCD and sent to the bank, which will decrypt  
 it and re-encrypt it using a key shared with the switch operator, such  
 as VISA. This *PIN translation* function is done entirely within an HSM.  
 VISA similarly uses an HSM to translate the PIN to a key shared with  
 the card-issuing bank, so it can be veriﬁed by an HSM there.

The ATM network rapidly became orders of magnitude bigger than Swift.

Rather than being used by a few thousand banks, it was soon connecting tens of  
 thousands of banks and hundreds of millions of cardholders. It was not feasible  
 to do either key exchange or ﬁnancial settlement bilaterally between 20,000  
 banks, so each bank connects to a switch provided by a switching organization  
 such as VISA, and these switches’ HSMs translate the traffic. The switches also  
 do accounting, so banks can settle their accounts for each day’s transactions  
 with a single debit or credit, rather than each having to maintain accounts with  
 thousands of other institutions.

The switches are trusted, so if something goes wrong there the consequences

can be severe. This seems to happen about once a decade. In one case a

switch manager ended up a fugitive from justice, and in another, 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. The bill in each case was in seven ﬁgures.

The engineers who designed ATM networks and security systems in the 1980s

(of whom I was one) assumed that criminals would be relatively sophisticated,  
 fairly well-informed about the system design, and rational in their choice of  
 attack methods. We worried about the many banks which were slow to buy  
 security modules. We worried about banks cutting corners such as omitting  
 authentication codes on authorization responses. We agonized over whether the  
 encryption algorithms were strong enough, whether the tamper-resistant HSMs  
 were tamper-resistant enough, and whether the random number generators used  
 to generate 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. Above all, we worried that a repairman would get his hands  
 on a bank’s PIN key, force the reissue of millions of cards and wreck public  
 conﬁdence in electronic banking. This was our doomsday scenario.

Doomsday eventually happened. In December 2017, a key at Postbank in

South Africa was compromised while kept on a laptop during during a data  
 centre move. Somehow, it was copied to a memory stick; the CEO also had  
 a copy. The copies were supposed to be destroyed in front of witnesses but  
 somehow a stick got lost. From March 2018 to December 2019, R56m (US

$3.4m) was stolen in 56,000 transactions, mostly from cards issued to poor

|  |  |  |
| --- | --- | --- |
|  |  |  |

pensioners to pay state beneﬁts. In February 2019, the central bank ordered  
 Postbank to reissue all its 12m cards, which cost R1bn (US $60m) [1237].

However, the millions of frauds against PIN-based payment cards over the

past 50 years turned out to be very much more diverse.

**12.4.2** **What goes wrong**

Card payment systems have huge transaction volumes, a wide diversity of oper-  
 ators, and plenty of capable motivated opponents. There have been successive  
 waves of card fraud, where vulnerabilities were discovered, exploited and then  
 eventually ﬁxed. The overall pattern is that card fraud has increased in value  
 over time but decreased as a proportion of the transactions; the system is slowly  
 getting more secure as it grows in both size and experience [91].

The ﬁrst wave, in the early 1990s, exploited the poor implementation and

management of early magnetic-strip card systems. In the UK, one proliﬁc fraud-  
 ster, Andrew Stone, was convicted three times of ATM fraud, the last time get-  
 ting ﬁve and a half years in prison. He started when he discovered by chance  
 an ‘encryption replacement’ trick: he changed the account number on his bank  
 card to his wife’s and found 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 wrote the encrypted PIN to the card’s  
 magnetic strip without linking it to the account number. 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 by arguing that the facts in each  
 case were different, and split it into thousands of small-claims cases that the  
 victims did not have the expertise to pursue. I was an expert in this case, and  
 used it to write a couple of papers on what went wrong [54, 55]. The fraud  
 eventually spread worldwide, as criminals in Romania and elsewhere started  
 designing ATM skimming equipment and sold it online. Here I’ll summarize the  
 more important and interesting lessons we learned.

Most of the actual ‘phantom withdrawals’ in the early 1990s appeared to

have one of the following three causes:

*•* Simple processing errors give rise to a steady background noise of dis-

month; that’s 240m a month in the UK alone. If the error rate is only 1  
 in 100,000, that’s a lot of disputes. Even if your core banking system has  
 good balancing controls, the peripheral systems that feed it can be ﬂaky.  
 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 bank’s server; periodically, the server itself

|  |  |  |
| --- | --- | --- |
|  |  |  |

crashed and forgot about open transactions, causing debits to be dupli-  
 cated. 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 reckoned in the 1990s to account for 30% of all  
 for years. For example, when I moved to Cambridge in February 1992 my  
 bank sent not one, but two, cards and PINs through the post, and they ar-  
 rived only a few days after intruders had got hold of our apartment block’s  
 mail and torn it up looking for valuables. In 2003–5, when magnetic-strip  
 cards were replaced with chip cards, there was another surge in thefts  
 from the mail – see ﬁgure 12.4. The main ﬁx was to make you phone a  
 call centre or visit a website to activate a card before you can use it.

*•* Frauds involving dishonest or negligent bank staff appeared to be the third  
 installing wiretaps inside an ATM to record customer card and PIN data,  
 and one case back in the 1990s of crooked insiders working out PINs for  
 stolen cards for £50 a time. More recently we’ve had bigger cases of crooks  
 working out how to social-engineer bank call centres to issue new cards  
 to addresses they control [2013]. Insider frauds were particularly com-

mon 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.

However, there were plenty of frauds due to careless design or that taught

technical security lessons.

*•* The shoulder-surﬁng trick of standing in an ATM queue, observing a cus-  
 blank card, 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 au-  
 tomated; 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 stan-  
 dard nowadays is to print only the last four digits of the account number  
 on the ticket, and since the early 1990s, cards have a three-digit *card ver-*  
 *iﬁcation value* (CVV) on the magnetic strip that must never be printed.  
 Yet the CVV is not always checked.

*•* There were many losses due to bugs and blunders. One ATM sold in  
 the 1980s had a ‘test dispense’ code that would output ten banknotes  
 of the lowest available denomination 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 dispense transaction. And despite the fact that I documented this in

|  |  |  |
| --- | --- | --- |
|  |  |  |

1993, and again in the ﬁrst edition of this book in 2001, similar incidents  
 were still reported as late as 2007.

*•* Some makes of ATM used in convenience stores could be reprogrammed  
 pensing twenties; it just took a default master password that was printed  
 in the online manuals. Any passer-by who knew this could stroll up to  
 the machine, reset the bill value, withdraw $400, and have their account  
 debited only $20. The store owners who leased the machines were not told  
 of the vulnerability, and were left to pick up the tab [1539].

*•* Many banks’ operational security procedures were dire. As an experiment,  
 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. (That bank later went bust in 2008.)

*•* One technique that’s worked consistently for 40 years – and still works  
 of tape, perhaps from an old videocassette, into the ATM throat and waits  
 for a victim. The card gets snagged in the loop, and the victim abandons  
 it. The crook retrieves it, and if he managed to see the victim’s PIN, goes  
 shopping. Some ATMs have mechanisms to frustrate this, and some don’t.  
 Some banks just don’t care: one victim of such a fraud, in a bank lobby,  
 went straight inside the bank to complain but was fobbed off by staff who  
 didn’t want to get involved. After her card was looted, her card-issuing  
 bank blamed her, and this ended up as a dispute.

*•* The high-tech modus operandi was using false terminals or skimmers to  
 there, crooks built a vending machine that would accept any card and PIN,  
 and dispense a pack of cigarettes. In 1993, two villains bought a real ATM  
 and a software development kit for it, programmed it to steal card data  
 and PINs, and installed it in the Buckland Hills Mall in Connecticut [988].

*•* False terminal attacks spread to Europe and to point-of-sale systems in the  
 was used to harvest card and PIN data in Utrecht, in the Netherlands;  
 and in 1994, crooks in London set up to a whole bogus bank branch [943].  
 Eventually, by the mid-2000s, card skimmers became widely available on  
 the black market. By 2015 a Romanian gang was caught operating 100  
 ATMs in tourist spots in Mexico, stealing $20m a month [1094]. Magnetic  
 strip cards were just too easy to copy, and the card technology had to  
 change.

*•* Since the mid-2010s, we have seen occasional ‘jackpotting’ attacks where  
 empty. This can involve infecting ATMs with malware, whether online

|  |  |  |
| --- | --- | --- |
|  |  |  |

or by getting physical access to a USB port, or physically inserting rogue  
 electronics [489].

*•* There are occasional frauds when an insider gets at one of the servers in  
 in customers being able to use cards with any PIN (if the online PIN  
 checking process fails) or in customers with the right PIN being able to  
 run up unlimited overdrafts (if the balance inquiry process fails). One  
 such failure was deliberate: after 9/11 damaged its ATM network, the  
 Municipal Credit Union decided to let customers in New York withdraw  
 money without checking their balances until things could be ﬁxed. That  
 cost $15m, and 118 customers ended up being charged with theft [1657].

I reckon the ﬁrst thing we did wrong when designing ATM security systems

in the 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 unable to use the security systems we designed. In recent  
 years, research by Yasemin Acar, Sascha Fahl and others has shown that many if  
 not most security failures can be seen as programmer usability failures; normal  
 programmers can’t cope with the complicated crypto APIs and access control  
 mechanisms that security geeks love to build [11]. Security geeks pay attention  
 to crypto because the maths are interesting, but less so to the ‘boring’ bits such  
 as creating tools that non-specialists can actually use. So it’s rare that the bad  
 guys have to break the crypto. And modern payment networks have so many  
 users that we must expect the chance discovery of vulnerabilities that were 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 eventually 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 [129, 216]. Since about 2005, skimmers made in Eastern  
 Europe are sold on underground markets, designed to be attached to the throats  
 of cash machines to read the magnetic strip and also capture the PIN using a  
 tiny camera or a keyboard overlay. I’ll discuss these in more detail in the next  
 section. The remedy has been moving from magnetic-strip cards to chip cards,  
 but this has taken over ﬁfteen years, and magnetic-strip fraud has cost a lot of  
 money in the meantime. The curious thing may be that it took 40 years from  
 the launch of magnetic-strip ATM cards until skimmers made them too easy to  
 attack. The key factor was that criminals started to specialise and organise, as  
 I discussed in section 2.3.

**12.4.3** **Incentives and injustices**

In the USA, the banks carry a lot of the risks associated with new technology.  
 In a historic case, Judd versus Citibank, 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 he “was  
 not prepared to go so far as to rule that when a credible witness is faced with

|  |  |  |
| --- | --- | --- |
|  |  |  |

the adverse ‘testimony’ of a machine, he is as a matter of law also faced with  
 an unmeetable burden of proof” – and gave her her money back [995]. The  
 US Federal Reserve incorporated this view into ‘Regulation E’, which requires  
 banks to refund all disputed transactions unless they can prove fraud by the  
 customer [639]. This has led to some minor abuse, but typically less than the  
 losses from vandalism [2046].

In other countries – such as the UK, the Netherlands and Norway – the

banks got away for years with claiming that their ATM systems were infallible.  
 Phantom withdrawals, they maintained, could not happen and a customer who  
 complained of one must be mistaken or lying. This position was somewhat

undermined in the UK when Stone and his followers started being jailed for  
 ATM fraud, and there were some rather unpleasant incidents. One example  
 was the Munden case [55].

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 account at the Halifax  
 Building Society empty. He asked for a statement, found six withdrawals for a  
 total of £460 which he did not recall making, and complained. The Halifax had  
 him prosecuted for attempting to obtain money by deception. It came out during  
 the trial that their IT was somewhat ramshackle; the disputed transactions had  
 not been properly investigated; and they made all sorts of wild claims, such as  
 that their ATM system couldn’t suffer from bugs as its software was written  
 in assembler. Nonetheless, it was his word against theirs. He was convicted  
 in February 1994 and suspended from the police force. Just before the appeal  
 was due to be heard, the prosecution served up a report from the Halifax’s  
 auditors claiming that their system was secure. The defense demanded equal  
 access to the bank’s systems for its own expert. The Halifax refused, so the  
 court disallowed all its computer evidence. The case collapsed, John Munden  
 was acquitted, and he got his job back.

Once the fuss died down, the banks went back to claiming that their systems

were secure, and 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. If a system

is to provide evidence, then dual control is not enough. It must be able to

withstand examination by hostile experts. The security property the bank really  
 needed wasn’t dual control but *non-repudiation*: the ability for the principals  
 in a transaction to prove afterwards what happened. This might have been

provided by installing ATM cameras; although these were mandatory in the  
 state of New York as an anti-mugging measure, 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 was a threat to the banks’ collective stance that their systems were  
 infallible. It would be a further 25 years before the Post Office case I mentioned  
 in section 12.2.6.1 would ﬁnally expose a bank’s systems to thorough scrutiny,  
 and have them condemned as unreliable in the High Court.

|  |  |  |
| --- | --- | --- |
|  |  |  |

**12.5** **Credit Cards**

The second component that led to modern card payment systems was the credit  
 card. For years after their invention by Diners Club in the 1950s, credit cards  
 were treated by most banks as a loss leader with which to attract high-value  
 customers. Eventually, the number of merchants and cardholders reached criti-  
 cal mass and the transaction volume took off. In Britain, from the mid-80s, the  
 credit card business was suddenly extremely proﬁtable14.

When you use a credit card to pay for a purchase in a store, the transaction

ﬂows from the merchant to their bank (the *acquiring bank*) which pays them  
 after deducting a *merchant discount* of typically just under 2% for a small mer-  
 chant15. If the card was issued by a different bank, the transaction now ﬂows  
 to a switch such as VISA which passes it to the *issuing bank* for payment. Each  
 transaction involves two components: *authorisation*, when you present your  
 card at a merchant and they want to know right now whether to give you the  
 goods, and *settlement*, which ﬂows through a separate system and gets money  
 to the merchant, often two or three days later. The issuer also gets a slice of  
 the merchant discount, but makes most of its money from extending credit to  
 cardholders.

**12.5.1** **Credit card fraud**

From the 1950s to the 1990s, credit card transactions were processed by mak-  
 ing a paper sales draft on a multipart form using the embossing on the card,  
 writing in the amount, getting the customer to sign it, and processing it like  
 a check. The risk of fraud using stolen credit cards was traditionally managed  
 by *hot card lists* and merchant *ﬂoor limits*. Each merchant got a local ‘hot

card list’ plus a limit set by their acquiring bank above which they have to call  
 for online authorization. In the 1980s, electronic terminals were introduced so  
 a sales clerk could swipe a card and get an authorization automatically. The  
 crooks’ response was a ﬂood of forged cards: between 1989 and 1992, magnetic  
 strip counterfeiting grew from an occasional nuisance into half the total fraud  
 losses [12].

The introduction of mail-order and telephone sales led to *card not present*

(CNP) transactions where the merchant was not able to inspect the card. Banks  
 managed the risk by using the expiry date as a password, lowering the ﬂoor lim-  
 its, increasing the merchant discount and insisting on delivery to a cardholder  
 address, of which the numerical part is supposed to be checked during autho-  
 rization. But the main change was to shift liability so that the merchant bore  
 the risk of disputes. If you challenge an online credit card transaction (or in  
 fact any transaction made under CNP rules), the full amount is immediately  
 debited back to the merchant, together with a signiﬁcant handling fee. This

14Payment systems have strong network externalities, just like communications technologies

or computer platforms: the service provider must recruit enough merchants to appeal to  
 cardholders, and vice versa, so new payment mechanisms can take years to get established,  
 then suddenly take off like a rocket.

15Debit cards are cheaper, and big merchants can pay under 1% even for credit card trans-

actions.

|  |  |  |
| --- | --- | --- |
|  |  |  |

applies whether the debit is a fraud, a dispute or a return.

VISA’s response to growing card forgery and online fraud was *card veriﬁ-*

*cation values* (CVVs) – 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: in the ﬁrst quarter of 1994, VISA’s fraud losses dropped  
 by 15.5% while Mastercard’s rose 67% [386]. So Mastercard adopted CVVs too.  
 They also appeared on debit cards, which converged with credit cards techni-  
 cally: this was an extended process as banks ﬁrst allowed credit cards to be used  
 in ATMs too and then let debit cards be used at the point of sale, at different  
 times in different countries.

The crooks moved to *skimming* – operating businesses where genuine cus-

tomer 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.  
 (In countries where PINs were already used in point-of-sale terminals, this al-  
 lowed forged cards to be used in ATMs directly.) The banks’ response was

intrusion detection systems that tried to identify criminal businesses by corre-  
 lating the purchase histories of customers who complained. By the late 1990’s,  
 the smarter crooked businesses learned to absorb the cost of the customer’s  
 transaction. You have a drink at a Maﬁa-owned bistro, offer a card, sign the  
 voucher, and fail to notice when the charge doesn’t appear on your bill. A month  
 or two later, there’s a huge bill for jewelry, electrical goods or even casino chips.  
 By then you’ve forgotten about the bistro, and the bank never had a record of  
 it [720].

In the early 2000s, high-tech criminals became better organised as electronic

crime became specialised. The emergence of online criminal forums, starting in  
 Russia and Ukraine in 2003, enabled malware writers, botnet herders, phishing  
 site operators and cash-out specialists to trade with each other and get good  
 at their jobs. This spilled over from targeting online transactions to attacks on  
 retail terminals. Forums offered fake terminals and skimmers that record mag-  
 strip card and PIN data, so as to make card clones. In the Far East, wiretaps  
 were used to harvest card data from the mid-2000s [1158].

Europe introduced smartcards in 2003–5, and the crooks came up with de-

vices that copy data from chip cards to mag-strip cards for use in terminals that  
 still accepted mag-strip transactions. Some of them used vulnerabilities in the  
 EMV protocol, and so I’ll come back to them after I’ve described EMV and chip  
 cards in the next section.

Regardless of whether the card has a chip or not, there are many scams

involving cards that are never received by genuine customers. There’s *pre-issue*  
 *fraud* including thefts from the mail of the ‘pre-approved’ cards that arrive in  
 junk mail. There are applications made in the names of people who exist but are  
 not aware of the application (often misrepresented as ‘identity theft’ by banks  
 that would like to pretend that it was your identity that was stolen rather than  
 their money [1324]). And there are scams where crooks get careless bank staff to  
 send a replacement card for your account to an address they control [2013]. The  
 remaining line of defence against such scams – until the customer gets a bill and  
 complains – is automatic fraud detection, which I’ll discuss in section 12.5.4.

|  |  |  |
| --- | --- | --- |
|  |  |  |

**12.5.2** **Online card fraud**

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 suspect got a list of hot credit card numbers from his partner who worked in  
 a supermarket, and used them to buy software from companies overseas, which  
 he downloaded to order for his customers. Hot card lists at the time carried  
 only those cards which were being abused in that country; using a local hot card  
 overseas meant that the bank would carry the can, not an innocent customer.  
 As it happens, the suspect quit before there was enough evidence to arrest him.  
 A rainstorm washed away the riverbank opposite his house and exposed a hide  
 the police had built to stake him out.

From about 1995, the dotcom boom got underway and businesses rushed to

build websites. There was anxiety 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 Mi-  
 crosoft and Netscape to introduce SSL/TLS to encrypt credit card transactions  
 en route from browsers to web servers.

The reality is a bit more complex. Intercepting email and web traffic is

indeed possible, especially at endpoints, but can be difficult to do at scale. Lots  
 of websites ran for many years with no encryption, or weak encryption, and  
 the real issue turned out to be not wiretapping but phishing. Even this only  
 got going at scale after 2004; and there (as I remarked in Chapter 3) the issue  
 is more psychology than cryptography. TLS per se doesn’t help, as bad guys  
 who can set up man-in-the-middle attacks can just get certiﬁcates and encrypt  
 the traffic. The site will have a different domain name, but it’s unreasonable  
 to expect most members of the public to notice that, especially as banks and  
 merchants use all sorts of variant domains themselves16.

Second, most of the credit card numbers that are traded online got into bad

hands because someone hacked a merchant’s computer. VISA had rules for years  
 that prohibited merchants from storing credit card data once the transaction  
 had been processed, but many merchants ignored them. There followed the

|  |
| --- |
| *Payment Card Industry Data Security Standard* (PCI-DSS), a joint e↵ort by the  Payment Card Industry Security Standards Council17. PCI DSS rules require  basic hygiene for systems holding cardholder data such as account numbers and expiry dates18 while sensitive data such as CVVs and PINs can’t be stored at all.  Finally, enforcement started to bite, and by in October 2007, the US National  Retail Federation asked credit card companies to stop forcing retailers to store  credit card data at all (they were supposed to store card numbers temporarily  in case of chargebacks) [1957]. PCI DSS has now become a signiﬁcant piece  of compliance for ﬁrms that accept credit card transactions; it provides little |

16There are now some technical ﬁxes, such as certiﬁcate transparency, which I’ll discuss in

section 21.5.1.

17This was set up by Visa, MasterCard, Amex, JCB and Discover; it now has other stake-

holders too.

18Cardholder data must be encrypted when they go over networks, and when stored they

must be protected by a ﬁrewall and AV; default passwords can’t be used; and you must  
 have a security policy, need-to-know access controls, testing, and since 2017 a secure software  
 development lifecycle. It adds up to quite a bundle of documentation and a lot of jobs for  
 accountants to check it.

|  |  |  |
| --- | --- | --- |
|  |  |  |

liability cover, since if fraud happens the banks can usually blame the merchant  
 anyway even if it was certiﬁed compliant.

Other real incentives facing merchants are, ﬁrst, the cost of disputes, and

second, security-breach disclosure laws. While the details differ between coun-  
 tries, disclosure laws have made a difference as notifying customers costs real  
 money and the 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 cus-  
 tomer fraudulently repudiated a transaction, the merchant would pursue them  
 through the courts. Nowadays many transactions are international, amounts  
 are small, and verifying overseas addresses via the credit card system is ﬂaky.  
 So the opportunity for repudiating transactions – and getting away with it – is  
 increased.

On the other hand, some market sectors have many websites that exploit

their customers, and porn sites have been a running sore. 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 [921]. 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 worst  
 case so far was probably Operation Ore, in which some three thousand victims  
 of credit card fraud were wrongly arrested on suspicion of buying child sex abuse  
 material, and at least one killed himself. I discuss the Operation Ore case in  
 section 26.5.3.

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 site are challenged by customers, your  
 margins will be eroded. If chargebacks go over perhaps 10%, your bank may  
 terminate your service. This has motivated merchants to take care – 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. Shared liability might well be  
 better, but legal systems are not good at that. One lobbyist beats another when  
 the law gets written, or one legal team beats the other when the key precedent  
 is set, and we get stuck with it.

One systematic attack involves progressive guessing. All websites must ask

for the primary account number and expiry date, but a merchant may also ask  
 for the CVV printed on the back of the card, and digits from the cardholder  
 address. Starting from a valid account number, you guess the expiry date by  
 testing it on merchant websites that check only that; then you guess the CVV  
 on websites that check that too, then the postcode digits, and ﬁnally guess the  
 house number from the websites that check that too. There are enough websites  
 out there for this to work for VISA cards; Mastercard has central monitoring,

|  |  |  |
| --- | --- | --- |
|  |  |  |

and they hot-list a number after about ten failed guesses (though this can lead  
 to denial-of-service attacks) [1].

Another attack is *credential stuffing*, where the bad guys get millions of

email / password combinations from compromised web sites and try them out in  
 other sites from which value can be extracted. Such attacks, plus the increasing  
 availability of stolen credit card data on underground markets, have driven the  
 development of better cardholder authentication, at least for larger transactions.

**12.5.3** **3DS**

3D Secure is a single sign-on system designed by the payment card industry19.  
 When the merchant captures a payment transaction past some threshold, they  
 redirect to a bank server that invites the customer to authenticate the transac-  
 tion using a password or a second-factor such as a code sent to their mobile by  
 SMS. It is increasingly used for large payment card transactions.

3DS acquired users rapidly because customers who used it were held liable for

fraud where possible, so merchants paid less. Customer onboarding was a soft  
 spot for years. Many banks initially let the 3DS servers enrol their customers  
 directly and solicit a password the ﬁrst time their card was used at a participat-  
 ing merchant, a process called *activation during shopping* (ADS). Some even let  
 customers re-enrol if they forgot the password, so initially the system was easy  
 to hack. It also got customers used to entering bank passwords at a site whose  
 URL has nothing to do with the bank, and one bank even got customers to enter  
 their ATM PINs there [1362]. Now, a decade after its initial roll-out, 3DS is  
 moving to an (incompatible) second version endorsed as an EMV standard. A  
 factor has been government mandates to use two-factor authentication, which  
 results in most banks knowing their customers’ mobile phone numbers. However  
 SMS-based two-factor authentication is now reaching the end of its useful life,  
 as discussed previously in section 3.4.1 and later in section 12.7.4. Some 3DS  
 implementations still use bank passwords.

**12.5.4** **Fraud engines**

People started working from the mid-1990s on better ﬁnancial intrusion detec-  
 tion, and by now all websites of any size that accept card-not-present trans-  
 actions have a fraud engine that decides whether to accept or decline each  
 transaction. There are two approaches: anomaly detection, which uses various  
 thresholding and other techniques to look for unusual patterns, and abuse de-  
 tection which looks for known fraud patterns. The big problem in both cases  
 is false positives. We all have experience of cards being blocked, and in many  
 cases the triggers are obvious. Small transactions used to cause alarms as they  
 suggested a thief testing stolen cards to see which are still live. Another issue  
 was multiple transactions overseas; in the 1990s, whenever I went to the USA,  
 my debit card would do three transactions and then stop working. Modern

machine-learning techniques have made such mechanisms slightly less annoy-

19It is variously branded as ‘Mastercard SecureCode’, ‘Veriﬁed by VISA’, ‘Amex SafeKey’

and ‘Discover ProtectBuy’.

|  |  |  |
| --- | --- | --- |
|  |  |  |

ing, but the sheer scale of modern payment systems with tens of thousands of  
 transactions per second means that even a 0.1% false positive rate will create a  
 ﬁrehose of customer complaints.

More convincing are projects that look for known patterns of misuse. For

example, FICO maintains a list of the most suspicious ATMs. Banks that

subscribe to its service tell it whenever a transaction is declined, whether because  
 of a stolen card, a wrong PIN or an empty account. The ATM is then bumped  
 up the ‘hot ATM’ list. When a crook takes a ﬁstful of stolen cards to an ATM,  
 it will get to the top of the list within three or four cards and then decline any  
 card issued by a bank that subscribes to FICO’s service. The crook will assume  
 they’re no good and throw them away. Over 40% of the world’s banks, by card  
 issuing volume, now subscribe.

An important success factor in running an intrusion detection system is the

incentives. Websites in the UK can turn away as much as 4% of offered shopping  
 baskets because of their fraud engines. If security is the responsibility of the  
 CFO, he’ll see it as a cost centre and try to minimise it; but for the chief  
 marketing officer, a 25% improvement in the false positive rate translates to  
 ‘1% more sales’, for which they’ll happily pay real money.

The core of a good fraud engine tends to be several dozen signals extracted

from the transaction stream on the basis of a set of well-understood threat  
 vectors (such as bad IP addresses, or too many logons from the same IP address)  
 and a set of quality signals (such as ‘card old but good’). These signals are  
 then fed to a machine-learning system that scores the transactions. The signals  
 appear to be the most important part of the design, not whether you use an  
 SVM or a Bayesian network. The signals need to be continuously curated and  
 updated as the bad guys learn new tricks, and the fraud engine needs to be  
 well integrated with the human processes. As for how fraud engines fail, the  
 regulator’s report into a 2016 fraud against Tesco Bank found that the staff  
 failed to ‘exercise due skill, care and diligence’ over the fraud detection rules,  
 and to ‘respond to the attack with sufficient rigor, skill and urgency’ [687]. In  
 that case, the bank failed to update its fraud engine following a warning from  
 Mastercard the previous day of a new type of card scam. We’ll discuss this case  
 further in section 12.6.3 once we’ve explained chip cards.

**12.6** **EMV payment cards**

The biggest investment since 2003 has been in new card technologies, with banks  
 replacing both credit cards and debit cards with EMV smartcards, followed by  
 contactless payments with both cards and phones. Card payments have become  
 both complex and diverse; the best way to understand them may be to follow  
 their evolution.

When integrated circuits came along in the 1960s and microprocessors in

the 1970s, various people proposed putting them in bank cards. The Germans  
 consider the smartcard to have been invented by Helmut Gr¨ottrup and J¨urgen  
 Dethloff in 1968, when they proposed and patented a custom IC for a card; the  
 Japanese point to a patent by Kunitaka Arimura in 1970; while the French credit

|  |  |  |
| --- | --- | --- |
|  |  |  |

Roland Moreno, who proposed 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.

Norway was second with some banks issuing chip cards from 1986. Britain’s

NatWest Bank developed the Mondex electronic purse system in the early 90s,  
 piloted it in Swindon, then sold it to Mastercard; the software evolved into  
 Multos, a card operating system that’s still in use. There was a patent ﬁght be-  
 tween VISA and Mastercard. There is more detail on these early pilot projects  
 in Chapter 3 of the second edition of this book. That was all good learning  
 experience. But for a payment card to be really useful, it has to work interna-  
 tionally – and especially in Europe with many small countries jammed up close  
 together, where millions of people cross borders for their weekly shop or even  
 on their commute to work. So the banks ﬁnally got together in the late 1990s  
 and hammered out a standard.

**12.6.1** **Chip cards**

The EMV standards specify chip cards and the supporting protocols for use in  
 ATMs and retail payment terminals. They were initially developed by Europay,  
 Mastercard and VISA, who then set up EMVCo to maintain and extend the  
 standards. Chip cards were rolled out in the UK from 2003–6 and then in other  
 European countries, most of which use PINs for authentication in stores as well  
 as ATMs, leading to the system being called ‘chip and PIN’. In the USA and  
 Singapore, chip cards are now used with signatures, and the system’s called  
 ‘chip and signature’. The standards run to many thousands of pages; they now  
 extend to contactless payments, online payments and much else; and there are  
 further documents speciﬁc to particular countries, and to individual banks. To  
 make sense of it all, let’s start with the basic protocol for using an EMV card  
 with a PIN to buy goods from a shop.

First, the card sends its credentials to the *PIN entry device* (PED) or termi-

nal, consisting of the primary account number (PAN) and a certiﬁcate signed  
 by the card issuing bank. Then the terminal sends an *unpredictable number*  
 or nonce *N*, the date *t* and the requested payment amount *X*, along with the  
 PIN entered by the cardholder. The card checks the PIN, and if it’s correct  
 it computes an *authentication request cryptogram* (ARQC) which is a message  
 authentication code (MAC) on *N*, *d*3 and *X*. Each message has some extra data  
 *di* which we’ll discuss later.

*C ffi! T* : *PAN, d*1*, CertKB*(*PAN, d*1)  
 *T ffi! C* : *N, t, X, d*2*, PIN*  
 *C ffi! T* : *d*3*, MACKCB*(*d*3*, T, N, t, X*)

The ARQC is computed using a key *KCB* shared between the card and

the bank20. The merchant can’t check this, so must either accept the risk of

20The long-term key *KCB* is actually used to generate a *derived unique key per transaction*

(DUKPT, pronounced duck-put) as a countermeasure to power analysis. I’m omitting such

|  |  |  |
| --- | --- | --- |
|  |  |  |

an offline payment or send the transaction to the card-issuing bank through  
 the payment network. The bank checks the ARQC and the available funds,  
 and if all’s well sends a response that also includes an *authorisation response*  
 *cryptogram* (ARPC) for the card. The card responds with a further MAC called  
 the *transaction certiﬁcate*.

EMV allows many options, some of which are dangerous, either individually

or in combination, and can be thought of as a construction kit for building  
 payment systems, with which you can build systems that are quite secure, or  
 very insecure. It’s the switch speciﬁcations from VISA and MasterCard that  
 really constrain the crypto as most banks want to be able to rely on their stand-  
 in processing. Things got tightened up steadily over 2005–17 as a succession of  
 frauds exploited the less secure versions. The simplest way to understand the  
 protocol suite may be to follow this history.

**12.6.1.1** **Static data authentication**

The default EMV variant up till 2011 in many countries was *static data authen-*  
 *tication* (SDA). As this used cheap cards that could not do public-key cryptog-  
 raphy, there’s no card public key *KC*, and the PIN is sent to the card in the  
 clear. So it’s still vulnerable to sniffing by a man-in-the-middle device, just as  
 with the magnetic strip cards that EMV was replacing. The terminal veriﬁes  
 the certiﬁcate and digital signature, but has no way to verify the MAC21. As  
 before, merchants have a ﬂoor limit below which offline transactions are permit-  
 ted, so they don’t have to stop trading when the network or the acquiring bank  
 are down22.

To begin with, the commonly-exploited vulnerability was backwards com-

patibility with magnetic strip cards. The certiﬁcate initially contained all the  
 information needed to forge a mag-strip card, and as the introduction of chip  
 and PIN meant that people started to enter PINs everywhere rather than just  
 at cash machines23, gangs either set up false terminals or used various wiretap  
 devices to collect card data from genuine terminals and then cashed out via  
 mag-strip forgeries. Initially these were used in local ATMs that would fall back  
 to mag-strip processing for reliability and compatibility during the changeover.  
 From the late 2000s, the crooks targeted countries such as the USA and Thai-  
 land that hadn’t adopted EMV yet. This wave of mag-strip fallback fraud is  
 visible in the yellow line in ﬁgure 12.4, which surges between 2006 and 2010.

Part of the crime wave of 2006–9 targeted petrol stations. An attack on our

local BP garage in Cambridge involved a CCTV camera ﬁtted in the ceiling  
 to capture the PINs plus a wiretap to get the card data; over 200 local people  
 found that copies of their cards were used in ATMs in Thailand. BP’s competitor  
 Shell was hit even harder, and fell back to mag-strip operation for a while after  
 some of their PIN pads were replaced with tampered ones by crooks pretending

details here and will discuss power analysis later in the chapter on side channels.

21The bank could thus use any algorithm it liked, but the default was DES-CBC-MAC with

triple-DES for the last block.

22Floor limits were ﬁrst cut to zero in Spain and this seems to be happening in the UK

too, which seems daft; stations should not stop selling tickets when the phone line goes down,  
 except possibly for season tickets.

23In the UK, at 900,000 shop terminals as well as 50,000 ATMs.

|  |  |  |
| --- | --- | --- |
|  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Losses (£m) | 5004003002001000 | Chip & PIN deployment period | | | 2006 | 580.7 | Phone banking | | | 452.7 | 2012 | Online banking | | | | 731.4 | 2018 |
| Card-not-present | | |
| Counterfeit | | |
| Lost and stolen | | |
| Mail non-receipt | | |
| ID theft | | | Mobile banking | | | |
| 2004 | | | 2008 | | 2010 | 2014 | | | 2016 |
| Total (£m) | 504.7 | 439.5 | 467.6 | 676.8 |  | 481.2 | 499.8 | 553.4 | 597.5 | 755.6 | 768.8 | 844.9 |
| Year | | |

Figure 12.4: – card fraud in the UK from 2004 to 2018.

to be maintenance engineers. The most spectacular fraud was discovered in

2008, when a gang apparently intercepted PIN entry devices in a warehouse  
 in Dubai, en route from the factory in China to the UK and the Netherlands,  
 and installed in them miniature mobile phones that sent the gang the card and  
 PIN data. Shops in the UK and banks in the Netherlands installed new devices  
 straight out of the box – which promptly started SMSing their customers’ data  
 to a server in Karachi [1729]. The gang was arrested and brought to trial in the  
 UK, but the case failed when the banks declined to provide evidence.

Colleagues and I therefore investigated a sample of PIN pads and found

that such attacks were easy. For example, the Ingenico i3300, the most widely-  
 deployed terminal in the UK in 2007, had a user-accessible compartment, shown  
 in Figure 12.5, which gives access to the bottom layer of the circuit board. We  
 found that a 1 mm diameter via, carrying the serial data signal, was easily  
 accessed using a bent paperclip, which could be inserted through a hole in the  
 plastic without leaving any external marks. So an attacker could indeed hide a  
 device inside the terminal that gathers and relays both card and PIN data. The  
 ‘Common Criteria Evaluation’ of such devices turned out to be worthless; I will  
 discuss the political and organisational reasons for its failure in section 28.2.7.2.  
 Such devices are now certiﬁed to standards set by PCI, and the rising issue  
 is software complexity; rather than being based on 8-bit microcontrollers, PIN  
 entry devices nowadays tend to be built on Linux or Android platforms, which  
 have a larger attack surface.

France was hit with a wave of attacks using ‘yescards’. These are cards

programmed to accept any PIN (hence the name) and to participate in the EMV  
 protocol using a certiﬁcate from a genuine card, but returning random values  
 for the MAC [179]. They worked just ﬁne to buy low-value items like snacks  
 and subway tickets, back when these were always sold via offline transactions.

|  |  |  |
| --- | --- | --- |
|  |  |  |

Figure 12.5: – 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.

Another family of problems has to do with authentication methods. Each

card, and each terminal, has a priority list of preferred *cardholder veriﬁcation*  
 *methods* (CVMs), which it shares in the supplementary data *d*1 and *d*2. The  
 card might say in effect: ‘ﬁrst try online PIN veriﬁcation, and if that’s not  
 supported use local PIN veriﬁcation, and if that’s not possible then a signature  
 will do, and if you can’t even get that then you don’t need to authenticate the  
 customer at all’. It might seem surprising that ‘no authentication’ is an option,  
 but it’s needed to support devices such as parking meters that don’t have PIN  
 pads. As well as PIN, signature or nothing, the terminal CVM list can specify  
 authentication on a device, such as the biometric scanner on a phone. Both card  
 and terminal can have risk-management logic to set monetary limits for different  
 methods. But EMV version 1 has a ﬂaw: the list of authentication methods isn’t  
 itself authenticated, so a crook can manipulate it in a false-terminal attack [169].

Many attacks become possible once you have a man-in-the-middle device.

Two students of ours implemented a *relay attack* for a TV programme; a bogus  
 terminal in a caf´e was hooked up via radio to a bogus card. When a journalist  
 in the caf´e went to pay £5 for some cake to a till operated by one student,  
 the transaction was relayed 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 [584]. There are many entertaining variants on the  
 theme. We don’t ﬁnd them in the wild, though, as they’re hard to scale.

The scale of fraud varies quite a lot between countries, and this teaches that

the practical security of EMV depends on contextual factors and implementation  
 details – such as the extent to which local ATMs will do fallback magnetic-  
 strip processing, the proportion of local shops open to various kinds of skimmer  
 attack, 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?

A landmark during EMV roll-out was the ‘liability shift’. In many countries,

regulators allowed banks to arm-twist merchants into installing EMV terminals  
 by changing their terms and conditions so that merchants were liable for dis-  
 puted transactions if EMV wasn’t used, but the banks became liable if it was.  
 In that case, banks in much of Europe simply blamed the customer: ‘Your card  
 was used, and so was your PIN, so you’re liable.’ So in theory fraud wasn’t  
 the bank’s problem any more. In practice fraud went up, as you can see from  
 ﬁgure 12.4. Fraud rose initially, thanks to the many cards stolen from the mail  
 during the changeover period; the banks rushed the roll-out as the merchants  
 paid for the fraud until they had EMV terminals, which took time24. There was  
 then a surge in counterfeit, as shops started to get terminals, people got used to  
 entering PINs in them, and the bad guys used bad terminals to steal card data  
 to make mag-strip copies for use in ATMs. The biggest change though was a  
 surge in mail order and online fraud. The net effect was that by October 2007  
 fraud was up 26% on the previous year [126].

The fraud ﬁgures would have been higher were it not for some blatant ma-

nipulation. UK bank customers were stopped from reporting card fraud to the  
 police from April 2007; this deal was negotiated between the banks and the  
 police by the Blair government in order to massage the crime statistics down-  
 wards, for which it was twice criticised by a parliamentary committee. Proper  
 fraud reporting was only reintroduced in 201525. You can see the effects of this  
 from the dip in the red line in ﬁgure 12.4 between 2008 and 2016; those missing  
 millions include a lot of fraud costs that were simply dumped on cardholders.  
 The banks also took over much of the ﬁnancing of the small police unit that  
 does investigate card fraud, so they have some control over such prosecutions  
 as do happen.

**12.6.1.2** **ICVVs, DDA and CDA**

In order to stop mag-strip fallback fraud, banks started from the mid-2000s to  
 implement the *integrated circuit card veriﬁcation value* (iCVV), a CVV that is

24This led to years of bad blood between merchants and banks.  
 25By then it had served its political purpose. From 2007–2015 crime fell steadily, as it was

moving online like everything else, and the online part wasn’t being counted properly. When  
 Theresa May stood for election as leader of the Conservative Party in 2016, one of her claims  
 to party members was that she’d cut crime despite cutting police numbers from 140,000 to  
 120,000; this claim was technically true, of reported crime at least. When Boris Johnson stood  
 to replace her in 2019, he claimed that crime had fallen while he was Mayor of London from  
 2008–16. This claim was not even technically true, as once the Office of National Statistics  
 insisted on counting properly from 2015, reported crime in Britain doubled.

|  |  |  |
| --- | --- | --- |
|  |  |  |

different in the card data in the chip from the versions on the magnetic strip  
 (which is read in mag-strip ATM transactions) and on the signature strip (which  
 is used online). Once all three are different, a chip-only skimmer can’t in theory  
 be used to make working mag-strip forgeries, and even if a merchant breaks  
 the PCI DSS rules by keeping the signature-strip CVV on a database that then  
 gets hacked, this CVV should not be enough to allow either a mag-strip forgery  
 or a yes-card forgery (this is known as *channel separation*). The three CVVs  
 are all calculated the same way – as a three-digit MAC on the PAN, version  
 number and expiry date, computed using triple-DES, but with different values  
 of a service code in the computation.

*Dynamic data authentication* (DDA) is the current default variant of EMV.

It was used initially in Germany and from 2011 throughout Europe. DDA cards  
 can do public-key cryptography: each has a private key *KC*, whose public key  
 is 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, assuring the terminal that the card is present (somewhere). The terminal  
 then sends a block containing the ‘unpredictable number’ and the PIN encrypted  
 using the card’s public key, followed by the transaction data, and the card  
 returns the application data cryptogram as before. This blocks skimmers from  
 collecting the PIN26. Back in the 2000s, DDA cards cost twice as much as SDA  
 cards; cards are now very much cheaper, and the main extra cost of DDA is  
 that card personalisation is slower.

*Combined data authentication* (CDA) is the Rolls-Royce variant. It’s like

DDA except that the card also computes a signature on the MAC. This enables  
 safer offline operation, as the terminal can now verify the transaction. It 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 the transaction data. As for why this matters, consider  
 the No-PIN attack.

**12.6.1.3** **The No-PIN attack**

In 2009, we got credible complaints from several fraud victims that their cards  
 had been stolen and then used in shops in transactions that their bank refused to  
 refund, claiming that their PIN had been used – while they insisted that it could  
 not have been compromised. Steven Murdoch, Saar Drimer and I investigated  
 and found that a man-in-the-middle device could tell the terminal that the card  
 had accepted the PIN, while telling the card that the terminal had initiated a  
 chip-and-signature transaction [1364]. Banks in some countries don’t use PINs,  
 typically because regulators didn’t allow the liability shift; and some banks in  
 the UK allow customers to refuse a PIN and get a US-style chip-and-signature  
 card instead.

In the protocol, the card data *d*3 contains a ﬂag indicating whether the PIN

was veriﬁed or not, and the terminal separately returns a ﬂag to its acquiring  
 bank with the same information. However the card ﬂag is proprietary to the

26As the card data are still in clear, a bad guy can still collect the PINs by visual observation

and try mag-strip fallback, in the hope that the card issuer doesn’t check CVVs; some banks  
 apparently still don’t.

|  |  |  |
| --- | --- | --- |
|  |  |  |

issuer, rather than an EMV standard, so it wasn’t checked by default.

Four criminals were arrested in France in May 2011, and a forensic report

was published by Houda Ferrari et al in 2015 after their last appeals ran out.  
 The No-PIN attack was accomplished by cutting out the chip from a stolen  
 card and bonding it underneath the chip of a hobbyist smartcard, which was  
 then programmed to perform the man-in-the-middle attack [680]. The gang

stole some e600,000 over 7,000 transactions using 40 modiﬁed cards, of which  
 25 were seized by the police.

One UK bank blocked the attack in late 2010 but the block was removed

in early 2011, perhaps because strict error handling was causing too many false  
 positives (the terminal ﬂag may be missing or wrong). The response to our dis-  
 closure of the vulnerability was somewhat negative; the banks’ trade association  
 wrote to the university asking it to take down the master’s thesis of a student  
 whose project had been to build a more robust man-in-the-middle device to in-  
 vestigate such issues (the university of course refused) [77]. It wasn’t until 2017  
 that the attack deﬁnitively stopped working in the UK. However, if either the  
 card or the merchant terminal was issued by a non-UK bank, the attack may  
 still work.

*Overlay smartcards* may have been used in China and possibly Italy for such

an attack in late 2018. These are very thin smartcards – about 180 microns thick  
 – with contacts top and bottom. They were developed in China to support  
 mobile phone roaming; the idea is that you stick one on top of your normal  
 phone SIM to provide an alternative. The overlay acts as a classic man-in-the-  
 middle. These devices are ideal for attacks; they’re widely available, they save  
 you having to build ﬁddly custom hardware, and are easy to use (you program  
 them in JavaCard).

**12.6.2** **The preplay attack**

On the 29th of June 2011, a Maltese customer of HSBC on holiday in Majorca  
 found four ATM transactions debited to his account despite the fact that he  
 had the card in his possession at the time. He’d eaten a meal the previous

evening at a restaurant where he thought the staff suspicious, and wondered  
 if his card had been copied. HSBC refused him a refund. So he contacted us  
 and we advised him to demand the transaction logs. It turned out that the  
 ‘unpredictable number’ generated by the ATM was just a 16-bit counter that  
 cycled every 3 minutes.

For a DDA/CDA card, the authentication step of EMV is:

|  |  |
| --- | --- |
| *T �! C* :  *C �! T* : | *T, N, t, X, d*2*, {PIN}KC*  *d*3*, MACKCB*(*d*3*, T, N, t, X*) |

If I know which ‘unpredictable number’ *N* a given terminal will generate

when the date *t* is tomorrow, and I have your card in my hand today, then I can  
 work out an ARQC *MACKCB*(*d*3*, T, N, t, X*) that will work tomorrow in that  
 machine. Mike Bond, Marios Choudary, Steven Murdoch, Sergei Skorobogatov  
 and I therefore instrumented a payment card, by attaching tiny microcontroller,

|  |  |  |
| --- | --- | --- |
|  |  |  |

memory and clock chips, and investigated ATMs around Cambridge, England.  
 We found that almost half of them used counters as ‘unpredictable numbers’.  
 Others had random number generators with stuck bits. We then went back to  
 the EMV specs and found that the test routine for a terminal only required the  
 tester to draw three ‘unpredictable numbers’ and check that they were different.  
 So could this be exploited at scale in Britain?

The next data point came in September 2012, when a Scottish sailor ordered

a drink in a bar in Las Ramblas, a tourist street in Barcelona. He paid e33 with  
 his EMV card, or so he thought. He passed out, woke up the following morning,  
 and found later that day that his account at Lloyds Bank had been hit with  
 ten debits of e3,300 each – a total of £24,000 at the time. The bank claimed  
 that as the chip and PIN had been used, he was liable. He instructed lawyers  
 who engaged us, and got the transaction logs from the bank. It turned out  
 that the ten transactions had been spaced evenly, ﬁled through three different  
 acquiring banks, and that although they had been made in the same terminal,  
 the terminal was registered with different characteristics at each of these banks.  
 This was clear evidence of technical manipulation, and the sailor got his money  
 back. We dubbed this the ‘pre-play attack’, as the essence is that rather than  
 replaying old transactions, you record transactions that you will book in the  
 future. If the same terminal will be used, then the fact that it’s the terminal  
 that generates the ‘unpredictable number’ makes the attack easy [282].

Since then, we’ve seen cases of pre-play attacks in a number of countries in

Europe, typically against customers of strip clubs and other sex industry ﬁrms.  
 In the UK, a customer of a lap-dancing club in Bournemouth complained in 2014  
 that the staff got him drunk and charged him £7,500 in 13 transactions [334].  
 Following press publicity, over a dozen other victims came forward, including  
 people who’d suffered debits after they were back home in bed [1948]. This  
 suggested a pre-play attack rather than a simple case of whores rolling drunken  
 customers; the local authority took an interest, and the club was put ‘on proba-  
 tion’ for six months. However we could not persuade the police to raid the club  
 and look for evidence, and eventually it got its full license back. In 2020, a club  
 in London actually lost its license after making multiple charges to customers,  
 with some victims being taken for tens of thousands [1341]. Elsewhere in Europe  
 too, it’s turned out to be hard; one such club in Cracow, Poland, got raided but  
 the police didn’t look for technical evidence. Terminals can be compromised in  
 various ways: apart from poor random number generators, their vendors can  
 fail to patch their software, and some nowadays even let operators run apps on  
 them27. So the preplay problem persists, and I fear that eventually we’ll have a  
 homicide case on our hands. Pimps who do pre-play attacks seem to often spike  
 the victim’s drink, and if you anaesthetise drunks and leave them to sleep it off  
 on a whorehouse sofa while you loot their bank accounts, then sooner or later  
 one of them will inhale some vomit.

An interesting point about security usability is that if you have four or ﬁve

cards in your wallet or purse, then if you add up all their balances and credit  
 limits, plus the extra ‘unauthorised overdrafts’ the card ﬁrms might give you,

27Dixons Carphone was ﬁned £500,000 in 2020 after malware infected 5,390 tills, compro-

mising the personal data of 14 million people and the data from 5.6 million cards. The previous  
 year they’d been ﬁned £400,000 for similar failures [2039].

|  |  |  |
| --- | --- | --- |
|  |  |  |

you’re probably walking around with the price of a car. If you had that much  
 cash in your pockets you’d probably not go into a bad part of town. You might  
 not even be comfortable walking along the high street unless you had a couple of  
 big friends with you. Payment cards obscure this prudential reﬂex, and enable  
 us to spend much more than we would when calm and sober. Quite apart from  
 fraud there are issues of vulnerability. The UK government, for example, has  
 just banned the use of credit cards in casinos. If you’re designing a system

that takes payments online for regulated products, or if your products might be  
 regulated in future because they can be addictive, then there’s a bunch of issues  
 you need to work through from ethics to geolocation to arbitration.

**12.6.3** **Contactless**

Contactless payment was pioneered in the USA by Mobil in 1997 and adopted  
 in the 2000s in a number of transport systems from London to Tokyo. By 2007,  
 you could just touch your phone on Japanese subway turnstiles in order to get  
 through. Barclays issued the ﬁrst contactless bank cards in the same year; VISA  
 and Mastercard developed contactless variants of EMV for payment; and Google  
 launched Android Pay in 2011 using the Mastercard PayPass standard28. These  
 early adopters struggled to get merchants to change their payment terminals,  
 while the press and public remained sceptical. The market tipped in 2014 when  
 Apple launched Apple Pay. By 2017 card payments had overtaken cash pay-  
 ments in the UK, because of the convenience of tap-and-pay; in 2018, debit cards  
 overtook cash in the USA, and the share of US consumers using mobile online  
 apps rose from 40% to 60% [707]. The coronavirus pandemic in 2020 caused a  
 further large-scale switch from cash to contactless, with UK ATM transactions  
 falling from 232m in January to 91m in April and cash transactions falling from  
 one in three to one in ten, while the contactless limit was raised from £30 to  
 £45.

The basic idea is simple. In the USA, the terminal generates an ‘unpre-

dictable number’ *N*, the card uses *KC* to generate a dynamic CVV as a 3-digit  
 MAC on selected transaction data, and this is sent to the card-issuing bank  
 along with *N*. In order to scale processing, the CVV keys may be made avail-  
 able to the HSMs of acquiring banks and to service ﬁrms that stand in for them.  
 Risk is mitigated by transaction limits – in 2020, $100 in the USA and £30 in  
 the UK. Some issuers have a policy that after a certain number of contactless  
 transactions, the cardholder must do a full EMV transaction with a PIN; this  
 causes complications in some applications. There’s a variant in the UK and  
 Europe where the card is made to generate an ARQC which may be sent to the  
 bank network for checking on a random basis.

As with regular EMV, *N* is generated by the terminal rather than by the

bank, so pre-play attacks are possible, but in most countries are not an issue  
 because of the transaction limits29. However, the extension of contactless pay-  
 ments from cards to phones led to additional complexity, and the systems we

28Full disclosure: I did some work for Google on the design.  
 29In Germany, you do high-value card payments by doing a contactless payment, combined

with online PIN veriﬁcation as in an ATM transaction, but I’m not aware of any pre-play  
 incidents.

|  |  |  |
| --- | --- | --- |
|  |  |  |

|  |
| --- |
| have now are a mash-up of competing proposals from the two card schemes.  In some Android phones, the credit card becomes a virtual credit card, im-  plemented in Java Card in a secure element in the NFC chip that does the  contactless RF protocol; Apple is something similar but with the key material  in the iPhone’s secure enclave. Other Android phones use *host card emulation*  where the NFC function is provided in software. NFC chips, or functionality, are  starting to appear in watches, bracelets and other devices too. Many use *tok-enization*, where the phone or other device is provisioned with a token30 and key  material by an online *tokenization service provider* (TSP) that acts on behalf of  the banks. The merchant sends the transaction to the TSP, which performs the  appropriate cryptographic operations in its HSM and forwards the transaction  to the customer’s bank. |

When contactless cards were rolled out, there were the usual implementation

failures. In some stores, you could be charged for a transaction twice if you paid  
 using a contact transaction yet left your wallet or purse near the terminal with  
 a different, contactless, card in it. Researchers also wondered whether a crook  
 could 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 in the mail, without opening the  
 envelopes [894]. Martin Emms and colleagues from Newcastle showed this was  
 possible, and found some even more interesting ﬂaws: one UK bank even let you  
 make one guess at a PIN; with others, the cash limit failed with foreign currency  
 transactions [628]. On November 5th 2016 this led to a major fraud against  
 Tesco Bank in the UK, when crooks in Brazil posted high-value transactions by  
 using mag-strip data on a contactless interface on a mobile device. The bogus  
 transactions amounted to £2.2m from 8,261 customer accounts and, although  
 the eventual losses were only £700,000, the attack created a ﬂood of fraud alerts  
 with which the bank’s weekend working procedures could not cope. It took

until November 7th to block the fraudulent transaction stream, many legitimate  
 transactions were also blocked, and normal customer service only restarted on  
 the 9th. For this failure, and the distress caused to customers, the regulator  
 ﬁned the bank £16.4m [687].

In 2019, Leigh-Anne Galloway and Tim Yunusov found you could increase

the contactless limit from £30 to £5500 by pretending to be a phone, and there’s  
 also an exploitable preplay attack. These attacks exploit the phone / card

/ terminal complexity. Android phones can have multiple limits depending on  
 whether the screen is off or on, and whether the user has recently authenticated;  
 and the phone and terminal send unauthenticated ﬂags to each other [736]. In  
 2020, David Basin, Ralf Sasse, and Jorge Toro found an improved middleperson  
 attack where a transaction is routed from a stolen card through two phones to a  
 contactless terminal, which accepts a claim that the cardholder was veriﬁed using  
 the phone’s own authentication mechanism, such as a biometric [182]. Possibly  
 such attacks could be prevented from scaling by the banks’ fraud engines, and  
 they haven’t appeared in the statistics (yet). However we still get complaints  
 from cardholders who have been victims of fraud after their cards were stolen,  
 and who claim their PIN wasn’t compromised while their bank claims it must  
 have been.

30There’s a *payment account reference* (PAR), a permanent pseudonym for the card number

|  |  |  |
| --- | --- | --- |
|  |  |  |

We’re starting to see innovative variants that don’t rely on speciﬁc hardware

but allow other channels to be used to run the protocol, such as QR codes. We’ll  
 have to wait and see whether these lead to man-in-the-middle attacks at scale.  
 The designers of second-generation EMV are talking of closing all the plaintext  
 gaps and even adding distance bounding as an option. Such techniques could  
 thwart many of the attacks described here. But the principal problems with  
 contactless now that it has been running for several years are more prosaic, and  
 include card collisions: if you have three cards in your wallet and you wave the  
 wallet over a subway turnstile, which of them gets debited? The card-choice  
 mechanisms aren’t robust enough to give repeatable answers [1287]. This is an  
 issue in London, where if you tap into the local transport system and fail to tap  
 out again, you get billed the maximum fare. If the entry and exit turnstiles see  
 different cards in your wallet, you end up paying double the maximum.

A recent development is *Software PIN on COTS* (SPoC) where the old as-

sumption of a sort-of cleartext magnetic strip plus a strongly encrypted PIN is  
 turned on its head: the SPoC rule is that devices where the PIN can’t be strongly  
 protected must never learn the associated card data. If a PIN is entered in a  
 merchant’s iPhone, as we now see at Apple stores, there’s another component  
 called a *Secure Card Reader – PIN* (SCRP) that plugs into the phone and ac-  
 cepts the customer card. Even if the phone app is compromised, the bad guy  
 doesn’t know which card the PIN will work for. The phone also passes the cus-  
 tomer PIN to the SCRP where it’s encrypted and sent off for online veriﬁcation.  
 There’s also work on ways to accept contactless payments on ordinary phones;  
 and presumably the next step will be to pay people by tapping phones together,  
 with one emulating the card and another the terminal. Direct phone-to-phone  
 payments are already routine for tens of millions of people in countries such as  
 Kenya and Bangladesh, as I’ll describe below in section 12.8.1. It will be an  
 interesting challenge to join up such systems with the world of EMV and make  
 the whole thing safe to use.

**12.7** **Online Banking**

After credit and debit cards, the third thread in the world of payments is banking  
 from your PC or phone.

In 1985, the ﬁrst home banking 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 previously in section 3.4.4.4 – it scared  
 the press and the bankers. But there was little real risk. The system allowed  
 only *nominated account payments* – you could only send money between your  
 own accounts and to accounts you’d notiﬁed to the bank, such as your gas and  
 electricity suppliers. In the early days this meant visiting a branch, ﬁlling a  
 paper consent form, and waiting until the cashier checked the payee account  
 number.

The early 1990s saw the rapid growth of phone banking, followed by bank

websites from the late 1990s, and then the phishermen arrived.

|  |  |  |
| --- | --- | --- |
|  |  |  |

**12.7.1** **Phishing**

In section 3.3.3 I summarised the history of phishing from its beginnings in the  
 1990s to its use against online bank accounts from 2003. The bad guys started  
 with crude lures from typosquatted domains like http://www.barqlays.com to  
 deceptive ones like http://www.barclays.othersite.com; the banks’ initial  
 response was to blame their customers. The gangs rapidly got more sophisti-  
 cated, as underground crime forums got going from around 2005 that supported  
 increasing specialisation, just as in the normal economy. One gang would write  
 the malware, another would herd the botnet, and we started to see specialists  
 who would accept hot money and launder it. The usual technique was to loot  
 whatever customer accounts you could and send the money to compromised  
 accounts at whatever bank was slowest at recovery. Of the £35m lost by UK  
 banks in 2006, over £33m was lost by a single bank. One of its competitors  
 told us that the secret was to spot account takeovers quickly and follow them  
 up aggressively; if money’s sent to a mule’s account, he should ﬁnd his account  
 frozen before he can walk to Western Union. So the laundrymen learned to  
 avoid them.

The industry learned to take down phishing websites as quickly as possible,

and specialist takedown companies got good at this. The bad guys responded  
 with tricks such as fast ﬂux, where phishing sites were hosted on botnets and  
 each mark who answered a lure was sent to a different IP address.

The second battleﬁeld was asset recovery: the fraudsters would try to get the

money overseas quickly and launder it, while the industry and law enforcement  
 would try to stop them. 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. After eGold got raided and closed  
 down by the FBI, the villains started to send money through banks in Finland  
 to their subsidiaries in the Baltic states and on to Russia. The third choice  
 was wire-transfer ﬁrms like Western Union: the phishermen recruit *mules* by  
 offering jobs in which they work from home and earn a commission as an agent  
 for a foreign company. They are told their work is to receive several payments  
 a week, deduct their own commission, and then send the balance onwards via  
 Western Union [789]. There have also been various electronic money services  
 in Russia and the Middle East [75]. Regulators played whack-a-mole: after one  
 channel got closed down, another would open up. Banks through which money  
 laundering was easy – known in the industry as ‘mule banks’ – even suffered less  
 fraud, as the big gangs avoided targeting their customers in the hope that they’d  
 stay useful for longer as the second link in the chain. This battle continues, with  
 funds laundered through everything from cryptocurrencies to Amazon gift cards.

This emphasises the importance of the *prevent – detect – recover* model we

introduced in section 12.2.4 above. Where authentication alone can’t do the  
 job, and you can’t ﬁnd other vulnerable points in the kill chain, you need to  
 beef up the intrusion-detection mechanisms that complement them.

|  |  |  |
| --- | --- | --- |
|  |  |  |

**12.7.2** **CAP**

|  |
| --- |
| In 2006, the banks announced a two-factor authentication standard based on  EMV, and this was launched the following year. The *Chip Authentication Pro-*  *gram* (CAP)31 consists of a handheld password calculator in which you can put  your EMV bank card. You enter a PIN; the device gets the card to check this;  you can then do one of three functions. You can get a one-time password to log  on, you can answer a logon challenge, or you can authenticate a series of digits,  typically from a payee account number and amount. |

Current versions use a custom app on the EMV card, which uses a key

shared with the issuing bank to compute a MAC on the supplied data and on  
 an *application transaction counter* (ATC) (which is different from the one used  
 for point-of-sale transactions). The response code is a truncated MAC and a  
 truncated ATC. The security is discussed in [585]; brieﬂy, if you put your card in  
 a bad terminal, this can generate a CAP code to log on to your online banking  
 service, though that’s hard to scale as you typically also need a password. The  
 availability of CAP readers means that a mugger who holds you up for your card  
 can demand your PIN and check it, without having to march you to an ATM  
 and risk being seen on CCTV. This has led to homicides, and was negligent  
 design: other password calculators just return the wrong result if you supply  
 the wrong PIN, including the early designs from the 1980s that I described in  
 section 4.3.2.

**12.7.3** **Banking malware**

As banks made simple phishing attacks harder by using ever more elaborate  
 authentication mechanisms from partial password questions to the early two-  
 factor authentication schemes, some bad guys just worked harder at persuasion.  
 Even in Germany, whose banks gave their customers printed lists of one-time  
 passwords, the crooks persuaded some customers to type them all in at once.  
 Other bad guys turned to automation, in the form of banking malware. From  
 2007, a series of malware strains such as Zeus, Torpig, SpyEye, EMotet, Trickbot  
 and Dridex stole hundreds of millions from banks and their customers worldwide,  
 spreading by various techniques including Word macros and drive-by downloads.  
 By 2011, man-in-the-middle attacks developed into *man-in-the-browser* attacks:  
 when the user of an infected PC sets out to use their bank account, browser  
 malware can actively modify transaction data so that what they see isn’t what  
 they authorise. This is why prudent banks now use a second factor such as CAP  
 to authenticate at least the last four digits of the account number of any new  
 payee. Banks who don’t use CAP may use a dedicated authentication device  
 instead, or a phone-based second factor.

**12.7.4** **Phones as second factors**

Another response to the wave of phishing in the mid-2000s was to use the  
 customer’s phone as a second factor. It seems natural to send a conﬁrmation,

31This is its brand name for Mastercard, which invented it; VISA calls it *Dynamic Passcode*

*Authentication* (DPA).

|  |  |  |
| --- | --- | --- |
|  |  |  |

such as: ‘If you really want to send $7500 to Russian Real Estate LLC, please  
 enter 4716 now in your browser.’ This appears to give the same beneﬁts as  
 CAP, but with a nicer user interface.

However, after South African banks started implementing this in 2007, they

quickly saw the ﬁrst *SIM swap* fraud. Some Johannesburg crooks got a new SIM  
 for the phone number of the CFO of a charity that looks after orphaned and  
 vulnerable children, and stole R90,460 from its bank account [1514]. The bank  
 complained to the phone company, which was unsympathetic: phone companies  
 sell minutes, not banking authentication services. As I discussed in section 3.4.1,  
 such frauds spread from South Africa to Nigeria, then to the USA from about  
 2014–5 where they were initially used to steal Instagram accounts, and from  
 2018 to loot people’s accounts at bitcoin exchanges [1092].

Such attacks now involve phone company insiders. In a 2019 case, an AT&T

contractor in Tucson, Arizona, helped a SIM-swap gang steal $2m from 29  
 victims [711]. In 2020, Kevin Lee and colleagues tried to swap ten SIMs on each  
 of ﬁve US phone companies and found it to be easy: with the big companies,  
 it worked every time. Vulnerabilities included authenticating people by asking  
 about recent calls and recent top-ups, both of which can be manipulated by an  
 attacker [1136]. It was also reported that SIM swappers were hacking phone  
 company staff, by social-engineering them into installing remote access tools  
 on their PCs, and then using the subverted machines to reassign target phone  
 numbers to SIMs they controlled [485]. Tens of thousands of customer service  
 reps are in a position to be careless, to get hacked or to take bribes from SIM  
 swap gangs. Some already take bribes to unlock stolen phones, and once these  
 underground communities link up we can expect things to get worse. There have  
 also been a couple of cases, in Germany and the UK, where attackers exploited  
 the SS7 signalling protocol to wiretap targets’ mobile phones remotely and steal  
 codes that way [489] (I’ll discuss this further in section 22.1.3). In China, the  
 law requires you to visit a phone shop and show ID to buy a SIM; in India,  
 you need a biometric check and the phone company is also made partly liable  
 for SIM-swap fraud. However, the direction of travel in the USA and Europe is  
 away from SMS as a second factor and towards a custom phone app32.

But as I wrote in the second edition of this book in 2007, “Two-channel

authentication relies for its security on the independence of the channels... if  
 everyone starts using an iPhone, or doing VoIP telephony over wireless access  
 points, then the assumption of independence breaks down.”

In the EU, the second payment services directive now requires banks to

use two-factor authentication. So it’s becoming universal, and the bad guys  
 are getting a lot of practice at breaking it. But what happens if you do your  
 banking not on your laptop but on a phone app, and use another phone app as  
 your second factor? If malware roots your phone, might it take over both apps,  
 and loot your account?

At the time of writing (2020), the European Central Bank takes the view

that two apps are OK so long as you use *runtime application self-protection*  
 (RASP), which means that you obfuscate the app code using the kind of tech-

32Data on which banks use hardware tokens as second factors, or software tokens, or SMS,

or no second factor at all, can be found at https://twofactorauth.org/#banking.

|  |  |  |
| --- | --- | --- |
|  |  |  |

niques developed during the 1980s for software copy protection and the 1990s for  
 digital rights management. This makes experienced security engineers wince,  
 as the history of such mechanisms is not a good one; it’s told in the chapter on  
 Copyright and DRM, and I discuss RASP further there in section 24.3.3. It is  
 very hard to get any assurance of how long an obfuscation scheme will take to  
 break; a break must be expected at any time, and the user of such a scheme  
 had better be ready to patch it immediately that happens. And maybe all an  
 attacker might need to do is shim one of the methods in the network stack to  
 get at the strings containing the authentication exchange. So they might not  
 need to extract the key or otherwise break the RASP mechanism itself.

**12.7.5** **Liability**

One long-running argument has been over liability. The rush to online banking  
 led many banks to adopt contract terms that put the risk of fraud on customers,  
 in conﬂict with consumer law and traditional banking practice [277]. Unfortu-  
 nately, the EU’s Payment Services Directives of 2007 and 2015 went along with  
 this by leaving a loophole in dispute resolution procedures33.

A study of the bank fraud reimbursement terms and conditions of 30 banks

operating in 25 countries showed a great variety of security advice, with much  
 of it being vague, impractical or even conﬂicting [201]. For example, HSBC  
 required unique PINs and passwords per account, contrary to advice given earlier  
 by the UK banks’ trade association which recommended customers to change all  
 their PINs to the PIN issued for one of their cards. It also had the most onerous  
 demands for Internet banking, including that the bank’s URL must always be  
 typed into the browser manually. It, and many other banks, required customers  
 to use antivirus software; fewer required that software be patched up-to-date.

Banks meanwhile trained their customers to be vulnerable by business prac-

tices such as telling their customers to reveal their security data, even when  
 making unsolicited calls. I’ve personally received an unsolicited call from my  
 bank saying ‘Hello, this is Lloyds TSB, can you tell me your mother’s maiden  
 name?’ You’re sorely tempted to tell them to get lost, but if you do it will be a  
 bother to reactivate or replace your payment cards. And even if the security rit-  
 ual is made more complicated, the phishermen can still talk the marks through  
 it, if need be as a man-in-the-middle (or browser) attack.

However, round about 2015, the bad guys started to evolve a better way.

**12.7.6** **Authorised push payment fraud**

*Authorised push payment* (APP) fraud refers to bank transfers that customers  
 are tricked into making. Figures only started to get collected in 2017 and the

33British banks got the UK government to insert ‘necessarily’ into article 72(2): ‘Where

a payment service user denies having authorised an executed payment transaction, the use  
 of a payment instrument recorded by the payment service provider, including the payment  
 initiation service provider as appropriate, shall in itself not necessarily be sufficient to prove  
 either that the payment transaction was authorised by the payer or that the payer acted  
 fraudulently or failed with intent or gross negligence to fulﬁl one or more of the obligations  
 under Article 69.’

|  |  |  |
| --- | --- | --- |
|  |  |  |

2018 ﬁgures are not calculated in the same way to 2017, so we don’t have those  
 on the graph in ﬁgure 12.4. However the total, at £354.3 million, is second only  
 to remote purchase fraud and more than the remainder put together.

A typical modus operandi is to look for someone who’s buying a house and

send an email that seems to be from their lawyer informing them that the ﬁrm’s  
 bank account number has changed. Another is to target vulnerable elderly

people. In one case, a 92-year old war veteran was called by crooks pretending  
 to be from his bank, bank A, who told him that the bank had been hacked, so  
 he had to transfer his life savings of £120,000 to bank B for safekeeping. Two  
 days later, his son visited and learned what had happened. In this particular  
 case, their lawyers demanded that bank B produce the know-your-customer  
 documents with which the mule account was opened. A few days later, bank B  
 (which had a reputation as a ‘mule bank’) sheepishly refunded the money.

That victim was lucky, but many were less so. Large frauds had become

easy because the banks had made large payments easy; in the old days, taking  
 out £120,000 would have involved arranging a meeting with a bank manager  
 at the very least. Yet online banking had been combined with a system of

instantaneous payments which meant that fraudsters could get away with ﬁve-  
 ﬁgure and even six-ﬁgure sums. In the UK this became such a sore point that  
 Parliament’s Treasury committee noted that rapid irrevocable payments were  
 simply the wrong default [1361], and the Payment Services Regulator changed  
 the rules so that the banks now carry some of the liability. As a result, it

has become signiﬁcantly more complicated to make large bank transfers. Even  
 medium-sized transactions get held up; if you try to pay your plumber a few  
 thousand for renovating your bathroom, you’re likely to get anxious calls from  
 the bank and be put through some security ceremonies.

Similar frauds have also been growing steadily against companies. Known as

*business email compromise* (BEC), they now account for several billion dollars  
 a year in losses [91]. In one recent case, a museum in the Netherlands agreed to  
 buy an 1855 painting by John Constable for £2.4m from a London art dealer,  
 but sent the money to the wrong account after crooks hacked the museum’s  
 email account and sent emails appearing to come from the dealer. The museum  
 sued the dealer but lost [506]. Victim ﬁrms have much less protection than

consumers do, but there are some mitigations that help both. For example,  
 the UK regulator ordered banks to implement *conﬁrmation of payee*: when you  
 ﬁrst make a payment to a new account, you’ll be asked for the account holder’s  
 name and you’ll be alerted if it’s wrong [1361]. Still, prudent practice is now  
 to hard-code company bank account numbers in business contracts, so that if  
 ﬁrm A pays crook C instead of ﬁrm B, there’s no room for argument over whose  
 fault it was. In Germany – where ﬁrms have been using direct bank payments  
 since the 20th century – it has been a legal requirement for years that companies  
 print their bank account numbers on their letterheads.

**12.8** **Nonbank payments**

There are many ways of making payments other than through banks. PayPal is  
 the survivor of a number of email-based payment service providers that sprung

|  |  |  |
| --- | --- | --- |
|  |  |  |

up at the time of the dotcom boom, and has now in effect grown into a bank, with  
 a portfolio of payment services both traditional and novel. A more traditional  
 service is *hawala*, a term that refers to money-changers that serve communities  
 of immigrants from South Asia and the Middle East, helping them to send  
 money home. They compete with Western Union, which grew up with the

Victorian telegraph network, and more modern payment service providers who  
 provide low-cost foreign exchange transactions. Some of these services are used  
 by cybercriminals, most notably PayPal and Western Union. Western Union  
 is a particular problem for law enforcement as criminals can send money to  
 any one of its many branches and withdraw it in cash. All such providers are  
 regulated in the European Union by the E-money directive of 2009, which sets  
 rules for capital and liquidity. There are also cryptocurrencies such as bitcoin,  
 which some regulators currently exempt from e-money regulation, and which  
 I’ll discuss in the chapter on Advanced Cryptographic Engineering.

Two particular types of payment service merit separate discussion: phone

payments and overlay payments, of which the leading examples are M-Pesa,  
 AliPay / WeChat Pay, and Sofort.

**12.8.1** **M-Pesa**

M-Pesa is a mobile phone banking service in Kenya, launched in 2007 by Voda-  
 fone. It took off rapidly and the ﬁrm that operates it, Safaricom, is now Kenya’s  
 largest ﬁnancial institution. Over 200 similar services have been launched in  
 less developed countries, and have been transformative in about 20 of them; the  
 largest such service now may be B-Kash in Bangladesh. Many such services  
 have been growing rapidly during the 2020 coronavirus lockdown.

M-Pesa got going as a means for migrant workers in Nairobi and Mombasa

to send money home to rural relatives. Before mobile phones came along, this  
 meant posting cash, or sending it with friends or bus drivers – both inconvenient  
 and risky, especially during a period of civil unrest in 2008 after a disputed  
 election the year before. Once mobile phones became widespread, people started  
 buying airtime as a means of transferring value, and from there it was a small  
 step to transfer actual value. The security mechanisms of such systems tend to  
 be simple, with an encrypted PIN, payee and value sent over SMS or USSD.  
 The key success factor is that phone companies have built networks of tens  
 of thousands of sales agents who can turn cash into digital credit and back  
 again – networks that reach the smallest villages, unlike the legacy banks. The  
 operational problems have to do with people sending money to the wrong phone  
 number by mistake, and integrating incoming M-Pesa payments with business  
 systems.

**12.8.2** **Other phone payment systems**

Many other countries have phone payment systems, or have widely-used pro-  
 prietary payment systems that work reasonably well on phones. An example  
 of these is PayPal, which redirects you from a merchant website to PayPal’s,  
 where you log in to authorise payment. Up until 2013, this was the world’s

|  |  |  |
| --- | --- | --- |
|  |  |  |

leading phone payment system. Since then the leading phone payment mecha-  
 nism has been AliPay, a proprietary payment app run by the Alibaba group in  
 China. It is closely followed by Tencent’s WeChat Pay; in 2020 they had 54%  
 and 39% of the Chinese mobile-payment market respectively. Smartphone pay-  
 ments took off rapidly in China, as M-Pesa did in Kenya, because banking used  
 to be unsatisfactory outside the main cities [608]. They have become the default  
 payment mechanism in China, and use a visual payment channel: a merchant  
 displays a QR code which the customer scans to send the right amount to the  
 right account. AliPay and WeChat Pay operate not just as business platforms  
 but as national infrastructure, and since 2018 are closely regulated: the Peo-  
 ple’s Bank of China gets copies of all transaction data [1529]. This ﬁts with the  
 Chinese approach to information sovereignty we discussed in section 2.2.2. And  
 both apps now support payment using your face, aligning with the growing use  
 in China of face recognition, a technology I discuss in section 17.3. India also  
 has a low-cost phone payment system in UPI, linked to the national Aadhaar  
 biometric card; on these national payment and identity layers sit a number of  
 competing payment apps.

**12.8.3** **Sofort, and open banking**

Credit cards were not traditionally used in Germany, which was inconvenient  
 when people started shopping online. One approach was to order goods from  
 a website, get the merchant’s bank account details and a transaction reference  
 number, go to your bank and pay, then go back to the merchant’s website the  
 next day and put in the payment details.

Sofort¨uberweisung is German for ‘immediate payment’ and set out to solve

this problem by means of an industrialised man-in-the-middle attack. In order  
 to buy a plane ticket, for example, you click the ‘sofort’ (‘immediate’) button on  
 the airline’s checkout page, and the service opens up a frame in which you enter  
 your bank name and account number. Sofort then logs on to your bank as you,  
 and presents you with the bank’s authentication challenge. Once you pass this,  
 it goes into your account, checks that there’s enough money, and sends itself the  
 payment. It then redirects back to the airline and you get your ticket [79]. The  
 effect is to make online shopping easier, but also to deprive the banks of card  
 transaction fees (the merchant pays about a third as much as they’d have paid  
 for a card transaction).

The banks sued Sofort for unfair competition and for inciting customers to

breach bank terms of service by entering their credentials at Sofort’s website.  
 They lost after the German Federal Antitrust Office argued that the banks’  
 terms of service hindered competition and were designed to exclude new business  
 models like Sofort’s. Sofort got a banking licence and the other banks just had  
 to compete.

The upshot was the EU’s second payment services directive (PSD2), also

known as ‘open banking’. Since January 2018, banks must open up their sys-  
 tems, not just by releasing transaction data in a standard format to other regu-  
 lated ﬁnancial institutions if their customer requests it, but allowing the other  
 institution to act as the customer does. The upside will include banks and ﬁn-  
 tech companies offering dashboards that will let you see all your holdings across

|  |  |  |
| --- | --- | --- |
|  |  |  |

all the banks with which you have an account, and move money between them  
 to get the best deals. The downside is that fraud and money laundering are  
 migrating rapidly to open banking channels. If a crook sets up an account at  
 Bank A, ﬁlls it with stolen money, authorises an account at Fintech B to op-  
 erate it, then uses B to get A to send money to C, A is not allowed to refuse  
 the transaction. The upshot is that traditional controls on fraud and money  
 laundering become much less effective. So there will be more jobs for security  
 engineers34. We will have to wait and see how all this develops.

The introduction of QR code based payment into the EMV standards opens

up the possibility of scaling something like Sofort’s payment mechanism world-  
 wide. As well as the customer presenting a payment instrument as a QR code,  
 the merchant can present a payment demand in this way, so that the customer’s  
 phone can initiate an online bank payment. Existing phone payment systems  
 like M-Pesa also require the customer to scan a QR code or enter data manu-  
 ally if their phone can’t do this. There may be some scope for innovation and  
 convergence here, so we’ll have to wait and see how it develops.

**12.9** **Summary**

Banking systems are critical to the security engineer because that’s how stolen  
 money gets moved – and fascinating in other ways too. Bookkeeping gives us  
 a mature example of systems oriented towards authenticity and accountability  
 rather than conﬁdentiality. The Clark-Wilson security policy provides a model  
 of this approach, which evolved over centuries. Making it work well in practice  
 means sophisticated functional separation, whose design involves input from  
 many disciplines. The threat model has a particular emphasis on insiders.

Payment systems played a signiﬁcant role in the development of cryptol-

ogy through their use in the ﬁrst generation of ATM systems; the adoption of  
 smartcard-based payments has changed the fraud landscape once more.

Finally, we have seen several waves of attacks on electronic banking systems

since the mid-2000s – by phishing account credentials, by man-in-the-browser  
 attacks by specialised malware, by SIM swap attacks on the mobile phones  
 used as a second authentication factor, and by social engineering customers to  
 send their money to the bad guys directly. These have progressively explored  
 the possible combinations of high tech and low cunning, and they teach the  
 importance of a holistic approach to fraud mitigation. The turbulence caused  
 by the pandemic is likely to emphasise this, but at least the mechanisms whose  
 use is surging, such as contactless payments in developed countries and phone  
 payments elsewhere, have had a few years to bed down.

34Open Banking means migrating from the old ISO 8583 standard to the newer ISO 20022.

This enables a move from 8-byte PIN blocks to 16-byte and thus from 3DES to AES; from  
 later batch settlement of transactions to real-time gross settlement; and much much more.

|  |  |  |
| --- | --- | --- |
|  |  |  |

**Research Problems**

I’ve always distrusted the cartel of big accountancy ﬁrms – down from the Big  
 Eight in the 1980s to the Big Four now, following three mergers and the failure  
 of Arthur Anderson in the Enron scandal. A student and I once wondered

whether being a client of big accountancy ﬁrm was a signal of wrongdoing, but  
 a brief analysis threw up no evidence either way. Thereafter when I served on  
 a governing body or audit committee I always proposed using a local ﬁrm, as it  
 was cheaper, but only once managed to get a change (and that was from one big  
 ﬁrm to another). When I served on our university’s governing body I had to put  
 up with this cartel shaking us down for a million a year and providing nothing  
 useful in return; most of the work was done by juniors. I thought the Germans  
 might be better off as their rules prevent auditors selling consultancy services,  
 but the Wirecard scandal punctured that illusion. The UK government still  
 decided after that scandal (and many others) that from 2024, the audit ﬁrms  
 must separate their audit and consulting practices in such a way that audit  
 partners’ remuneration comes only from the audit business and is not cross-  
 subsidised from consultancy [1050]. It would be great if that works, but I fail to  
 see how it can have any real effect on most of the concrete problems described  
 in this book, whether the internal control issues analysed in this chapter or the  
 assurance issues which I tackle in section 28.1. The audit cartel imposes huge  
 social costs and is not quite what we expect from standard economic analysis35.  
 It needs to be understood better.

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. Just as many security  
 failures come from poor usability at the level of both users (who are offered  
 dangerous choices as defaults) and programmers (who’re given access-control  
 and other tools that are insanely tricky to use), so many internal control failures  
 come from administrative mechanisms that are designed for the comfort of the  
 auditor rather than to be actually usable in real organisations. How can we do  
 this better?

Payment systems are at the one time deeply conservative, being in many

ways little changed since the 1970s, and also constantly evolving, as the mecha-  
 nisms moved from ATMs and HSMs to chip cards and to crypto chips in mobile  
 phones. The ground’s also shifting as attacks evolve (as with SIM swap) and  
 the environment changes (as with open banking). Maintaining resilience in the  
 face of such change takes work. As EMV implementations get tightened up, and  
 as the second version of EMV starts to tackle the residual vulnerabilities de-  
 scribed here, we can expect fraud to move to the periphery: to the customer, via  
 account takeover; to the merchant, via hacking attacks, refund scams, coupon  
 scams and the like; and to the bank, via preissue frauds and technical attacks  
 on the systems for authorisation and settlement.

If account takeover is going to become ever-more pervasive, what are the

implications? I suspect that our regulatory approach needs an overhaul: blam-

35See the Lerner-Tirole model discussed in section 28.2.8 for a model of how ﬁrms faced

with a compliance requirement usually choose the cheapest supplier. Why do most large ﬁrms  
 and even large universities go for famous but expensive ﬁrms when they’re all but useless at  
 detecting whether executives are crooks or ﬁrms are trading while insolvent?

|  |  |  |
| --- | --- | --- |
|  |  |  |

ing ordinary customers for harm they suffer from systems designed by others  
 is wrong. But what should we do? Should we go for radical transparency, im-  
 pose payment delays, or put more weight on rapid asset recovery? Is there some  
 smart combination, such as making the speed and ﬁnality of payment a function  
 of the known standing of both payer and payee? Or should regulators just keep  
 pushing liability back to the banks and let them work it out?

The context in early 2020 was that retail banks are making less money than

they used to, because of low interest rates and growing competition, so bank  
 security engineers are being asked to do more with less. Social media are making  
 downtime more painful; if a bank’s mobile app is down for 15 minutes because of  
 a DDoS attack on a gateway, there can be a twitter storm that causes directors  
 to phone the Chief Operating Officer. Such incentives push in the direction of  
 moving stuff to the cloud, but this raises further problems; we’ll discuss cloud  
 HSMs later in the chapter on Advanced Cryptographic Engineering. The coro-  
 navirus pandemic has been great for payment service providers, with PayPal’s  
 share price up by about a half; as to where it may drive ﬁntech innovation, per-  
 haps it will be around video. Videoconferencing is having to replace in-branch  
 meetings for complex and high-value transactions such as loans. The latest wave  
 of ﬁntechs such as Monzo were already getting customers to record a selﬁe video  
 as part of the onboarding process, so that call centre staff helping a customer  
 recover an account from a lost or stolen phone could check they’re the same  
 person who opened the account. What else?

**Further Reading**

Andrew Jamieson wrote a 100-page ebook on EMV for Underwriters’ Labora-  
 tories – ten times what I had space for here [977] – and that may be a useful  
 stepping stone from my short summary to the thousands of pages of speciﬁ-  
 cations from PCI SSC and EMVco [629]. I don’t know of any comprehensive  
 book on core banking systems, although there are many papers on payment  
 systems available from the Bank for International Settlements: the most re-  
 cent, as we go to press in July 2020, analyses quality of service and notes that  
 while payments within Europe mostly take under 30 minutes, a combination of  
 multiple intermediaries, business hours, time zones, capital controls, liquidity  
 and ancient technology mean that payments to Asia and Africa can take hours  
 to days [162]. If you’re going to do any real work on internal control, you’d  
 better read ISA 315 [950]; its interpretation by the big four accountancy ﬁrms  
 now makes the weather on internal controls. I’ll revisit this topic in Part 3.  
 To understand what can actually go wrong, read the judgment in the Horizon  
 case [185] and the survey of corporate fraud by Alexander Dyck, Adair Morse  
 and Luigi Zingales [596].

The IBM system of generating and protecting ATM PINs was described in

a number of articles, such as [521] and [951], while early ATM networks are  
 described in [763]. For the basics of ATM fraud, see [55]; while the transcript  
 of the trial of an HSBC insider gives a snapshot of typical internal controls in  
 electronic banking systems [1569]. The ﬁrst survey of underground markets was  
 2007 by Jason Franklin, Vern Paxson, Adrian Perrig and Stefan Savage [714];

|  |  |  |
| --- | --- | --- |
|  |  |  |

even then, the focus was on bank fraud rather than on drugs or malware. There’s  
 a rich literature since then on topics from the social dynamics of underground  
 communities [1345] to the Russians behind the Dridex malware campaign [1622].  
 Colleagues and I have contributed to big surveys of cybercrime in 2012 [90] and  
 2019 [91]. There’s a collection of our group’s writings on bank fraud at our Bank  
 Fraud Resource Page, at https://www.cl.cam.ac.uk/~rja14/banksec.html.  
 For an authoritative case study of a large card fraud, see the FCA’s 2018 ruling  
 against Tesco Bank [687]. This not only sets out how the fraud was done, but  
 how the controls failed at multiple points and how the regulators calculated the  
 ﬁne.

Finally, for the political and legislative history of the US intelligence initiative

against terrorist ﬁnance and its efforts to get SWIFT data by covert or legislative  
 means, see David Bulloch’s thesis [341].

|  |  |  |
| --- | --- | --- |
|  |  |  |