The Correctness of Crypto Transaction Sets (Discussion)

Ross Anderson

Cambridge University

This talk follows on more from the talks by Larry Paulson and Giampaolo Bella that we had earlier. The problem I'm going to discuss is, what's the next problem to tackle once we've done crypto protocols? We keep on saying that crypto-protocols appear to be "done" and then some new application comes along to give us more targets to work on – multi-media, escrow, you name it. But sooner or later, it seems reasonable to assume, crypto will be done. What's the next thing to do?

The argument I'm going to make is that we now have to start looking at the interface between crypto and tamper-resistance.

Why do people use tamper resistance? I'm more or less (although not quite) excluding the implementation of tamper resistance that simply has a server sitting in a vault. Although that's functionally equivalent to many more portable kinds of tamper resistance, and although it's the traditional kind of tamper resistance in banking, it's got some extra syntax which becomes most clear when we consider the Regulation of Investigatory Powers (RIP) Bill. When people armed with decryption notices are going to be able to descend on your staff, grab keys, and forbid your staff from telling you, then having these staff working in a Tempest vault doesn't give the necessary protection.

1 Cryptography Involving Mutually Mistrustful Principals

In order to deploy "RIP-stop cryptography" (the phrase we've been using), you need to get guarantees which are mandatory – in the sense of mandatory access control. That is, you need guarantees, that are enforced independently of user action and are therefore subpoena proof, that a key was destroyed at date X, or that a key is not usable on packets over a week old, or that a key is only usable in packets over a week old if it's used with the thumb print of the corporate chief security manager, or whatever. You could do this with something like Kerberos, you just say that tickets can only be decrypted (as opposed to used) if the corporate security manager says so (this is maybe the cheap and cheerful way of getting RIP-stop into Windows). Alternatively, you could impose a requirement that a key should be unusable in the same key packet twice – possible if you've got a device like the 4758 with a reasonable amount of memory – or you could have a key which is unusable without a correct biometric. All of these give you ways of defeating RIP.

B. Christianson et al. (Eds.): Security Protocols, LNCS 2133, pp. 128–141, 2001.

Now when we look at the larger canvas, at the applications that really use hardware tamper-resistance, we find that most of them are used where there is mutual mistrust. Often all the principals in a system mistrust each other.

The classic example, that Simmons discussed in the 80's, was the business of nuclear treaty verification, and indeed command and control generally [10]. The Americans don't trust the Russians, and the Pentagon doesn't trust the commanders in the field not to use the weapons, and so you end up with this enormous hierarchy of tamper-resistant boxes.

Pre-payment electricity meters provide another application that we have discussed at previous workshops [3]. There you have a central electricity authority, and you have hundreds of regional electricity vendors who sell tokens which operate meters. How do you balance power and cash, and stop the various vendors running off with the cash – or selling tokens that they then don't own up to? The only known solution is using some kind of hardware tamper-resistance, at least in the vending stations.

Then you've got bank security modules, which are basically used because banks don't want to trust each others' programmers not to get at customer PINs. Matt asked yesterday: "How could you possibly use tamper-resistance where you have got mutual mistrust?" In practice, tamper-resistance is used precisely where there is mutual mistrust, and this is what makes it difficult and interesting.

A brief history of tamper-resistance includes weighted code-books, sigaba cypher machines with thermite charges, and so on – all fairly familiar. GSM is fairly low-level stuff: your SIM contains a customer key K_c, and this key is *you*, for practical purposes, in the network; it is used to respond to random authentication challenges. You have no real motive to break into the SIM; it's just convenient to use a smartcard to hold K_c (although it does prevent some kinds of cloning such as cloning a phone in a rental car).

Pay-TV becomes more serious, and Markus Kuhn has talked about it at some length [4]. There the mutual mistrust is between the pay-TV operator and the customer. The customer has the master key on his premises, and the pay-TV operator doesn't trust the customer not to use the master key. Hence the need for tamper-resistance.

Banking is an issue that I've written about a lot [2], and the kind of security modules that I worked with (and indeed built some of) are basically PCs in steel cases with lid switches. Open the lid, and bang goes the memory. This is a very simple way of building stuff. You can make it more complex by putting in seismometers, and photo-diodes, and all sorts of other sensors, but the basic idea is simple. When you take it to its logical conclusion you get the IBM 4758, where you have got a DES chip, microprocessor, battery and so on, potted in a tamper-sensing core with alarm circuits around it.

What are the vulnerabilities of such products?

Many of the kinds of vulnerability we had in the 80's were to do with maintenance access. The maintenance engineer would go in to change the battery; he could then disable the tamper-sensing; and he could then go back on his next visit and take out the keys. That got fixed. The way to fix it was to take the batteries outside the device, as you can see in the photo of the 4758 in figure 1. Then there is nothing user-serviceable inside the device, so as soon as you penetrate the tamper-sensing membrane the thing dies irrevocably – it becomes a door-stop.

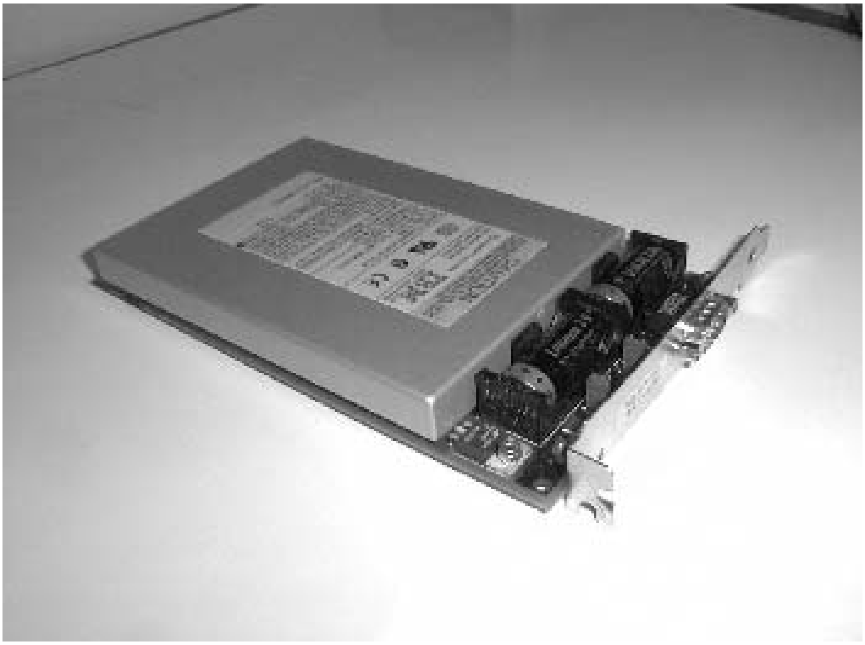

Fig. 1. – The IBM 4758 cryptoprocessor (courtesy of Steve Weingart)

We've more or less got to the point that the hardware can be trusted. We deal with the problem of whether the software can be trusted by getting FIPS 140-1 evaluation done by independent laboratories that sign a non-disclosure agreement and then go through the source code. That may be enough in some cases and not enough in others, but that isn't the subject of this talk.

2 Cryptoprocessor Transaction Sets

What I'm interested in is the command set that the cryptographic device itself uses. We have up till now, in this community, been looking at what happens in protocols where Alice and Bob exchange messages, and you need to contemplate only three or four possible types of message.

In the real world, things are much harder. If Alice and Bob are using tamper-resistant hardware, then they have got boxes that may support dozens or even hundreds of transactions. And we are assuming that Alice and Bob are untrustworthy – perhaps not all the time, perhaps you're just worried about a virus taking over Alice's PC – but perhaps Alice is simultaneously your customer and your enemy, in which case you can expect a more or less continuous threat.

2.1 Early Transaction Sets – CUSP

Let me review quickly how cryptographic processor instruction sets developed. The first one that's widely documented is IBM's CUSP, which came in in 1980 in the PCF product, and shortly after that in the 3848 [8]. This was a device the size of a water-softener that hung off your mainframe on a channel cable. It was designed to do things like bulk file encryption, using keys that were protected in tamper-sensing memory.

What IBM wanted to do was to provide some useful protection for these keys. If you merely have a device which will, on demand, encrypt or decrypt with a key that is held in its memory, then it doesn't seem to do you much good. Anybody who can access the device can do the encrypting and the decrypting.

So there began to be a realization that we want, in some way or another, to limit what various keys can be used for. Although the details are slightly more complex than the subset I'm giving here, the idea that IBM had was that you can divide keys into different types, and some keys can be local keys and other keys can be remote keys. How they did this was to have three master keys for the device – basically one master key and two offsets that would be XORed with it to create dependent master-keys. A key that was encrypted with the main master key, you can think of that as a blue key, and a key that was encrypted with variant KMH_0 you might think of as a green key, and a key that was encrypted with variant KMH_1 you might think of as a red key.

Now few people go through this documentation [7], because the IBM terminology and notation in it are absolutely horrible. But by starting to think in terms of key typing, or key colours, you can start to make much progress.

One of the goals that IBM tried to achieve with the 3848 was to protect for host-terminal communications using a session key which is never available in the clear, and which therefore has no street value. (This was an NSA concern at the time, because of their various staff members who had been selling key material to the Russians.) So how do you protect communication between a mainframe and its terminal?

The implementation was that you had a host transaction ECPH, which would take a green key – that is a session key encyphered under KMH0 – and a message m, and it would, in the trusted hardware, get its hands on this green key by decyphering it under KMH0, and then apply it to encypher the message m.

ECPH: $\{KS\}_{KMHO}, m \longrightarrow \{m\}_{KS}$

So you had a means of using the green key to encypher messages, and 'can encypher with green key' is a property of CUSP. Similarly, you can decypher with a green key, because can supply K_S encyphered under KMH_0 and a message encyphered under K_S and you will get back m.

How do you go about generating keys?

The technique is to supply a random number, or a serial number – or any value that you like, in fact – and use the device to decypher that under KMH0 to get a green key. In other words, the external encyphered keys that you hold on your database are just numbers that you thought up somehow. They are then 'decyphered' and used as keys.

Then you've got a more sensitive transaction, 'reformat to master key' (RFMK), which will take a master-key for a terminal – which is set up by an out-of-band technique – and K_S encrypted under KMH0, and it will encypher the session key under the master-key for the terminal.

RFMK: $\{KS\}_{KMHO}, KMT \longrightarrow \{KS\}_{KMT}$

So we've got another type of key, a red key I've called it, which is assumed to be held in the terminal. At the terminal, which is a different piece of hardware with different syntax, you've got a transaction DMK which will take a session key encyphered under KMT, and the clear value of KMT which is held in the hardware; it will give you a value of K_S which can then be used to do the decryption. (There isn't any hardware protection on the syntax of crypto in the terminal, because it is assumed that if you have physical access to the terminal then you can use the keys in it.)

Now this isn't all of the implementation, because there are other things that you need to do. You need to be able to manage keys, and (most confusingly of all) you need to be able to manage peer-to-peer communication between mainframes, which was being brought in at the time. So you end up having transactions to set up a key that is a local key at one mainframe and a remote key at the other mainframe. The IBM view at the time was that this made public-key cryptography unnecessary, because DES was so much faster and you needed tamper-resistant hardware anyway, so whay not just have local keys and remote keys? They have got the same functionality.

What went wrong with this? Well, it was very difficult to understand or explain, because the terminology they used is very complex. All sorts of implementation errors resulted. People found that it was simpler just to use blue keys which are all-powerful, rather than to mess around trying to separate keys into red and green and then finding that various things they wanted to do couldn't be done.

Nobody really figured out how to use this CUSP system to protect PINs properly as they were being sent from one bank to another. For example, a PIN from another bank's customer would come in from your cash machine, so you would decypher it and then re-encypher it with a key you shared with VISA for onward transmission. There were one or two hacks with which people tried to prevent PINs being available in clear in the host mainframe – such as treating PINs as keys – but for various reasons they didn't really work.

2.2 Banking Security Modules

So the next generation of security hardware to come along. The VISA security module, for example, takes the concept a bit further. They've actually got about five or six different colours of key.

The basic idea is that the security module is generating the PIN for a cash machine, by encrypting the account number using a key known as the PIN key. The result is then decimalised, and either given to the customer directly or added to an 'offset' on the customer's card to give the PIN that they must actually enter:

Account number:	8807012345691715
PIN key:	FEFEFEFEFEFEFEFE
Result of DES:	A2CE126C69AEC82D
Result decimalised:	0224126269042823
Natural PIN:	0224
Offset:	6565
Customer PIN:	6789

So you start off with keys which *never* decrypt, such as the PIN key KP, and you also have keys which *can* decrypt. Now the PIN key is a 'red key', a key which can never decrypt, and keys of this type are also used to protect PINs locally. (For example, when a PIN is encrypted at a cash machine for central verification, then the key used to perform this encryption is treated as a red key by the security module.) You can pass the security module a PIN which has been encrypted under a red key and it will say whether the PIN was right or not. The green key operates like a normal crypto key; you use it for computing MACs on messages and stuff like that.

So you use a red key to compute the PIN from the primary account number (PAN), and you use a red key to encipher the PIN from the ATM to the host security module. The whole thing works because nothing that's ever been encrypted under a red key is supposed to leak out – except the fraction of a bit per time that comes out from the PIN verification step. Then you add various support transactions, like 'generate terminal master key component', 'XOR two terminal master key components together to get a terminal master key' and so on and so forth.

What's the security policy enforced by the set of transactions that the security module provides? Well, it doesn't quite do Bell-LaPadula, because if you think of the red keys as being High, it allows an information flow from High to Low – about whether a PIN at High is right or not. So you need an external mechanism to track how many PIN retries have there been on an account, and if there's been more than three or 12 or whatever your limit is, then you freeze the account.

It doesn't quite do Clark-Wilson either, because you can generate master key components and XOR them together, and there's no system level protection for separation of duties on these two master key components; that's part of manual procedure. But the security module is moving somewhat in the direction of Bell-LaPadula and Clark-Wilson (although its evolution went down a different path).

Once you network many banks together, you've got further requirements. You've got one type of key used to encrypt a PIN on the link between the cash machine and the security module of the acquiring bank; you've got another type of key being used between the acquiring bank and VISA; another type of key being used between VISA and the issuing bank; then you've got the top level master keys which VISA shares with banks, and so on. You can think of these as being all of different colours.

So you've got all these various types of key, and you've got more and more complex transactions; the VISA security module now has about 60 different transactions in its transaction set. You've got support transactions, such as translating different PIN formats. You also have got potential hacks which I will come to later.

So the concern with something like the VISA box is: "Is there some combination of these 60-odd transactions, which if issued in the right order will actually spit out a clear-text PIN?" The designers' goal was that there were supposed to be one or two – in order to generate PINs for customers and keys for ATMs, for example – but they all involve trusted transactions that involve entering a supervisor password or turning a metal key in a lock. So the issue of trusted subjects is supposedly tied down.

2.3 The IBM 4758 Security Module Product

Now we come to the more modern IBM product, the 4758. There's another picture of a 4758 in figure 2 with the shielding slightly removed. You can see a protective mesh here, a circuit board inside it behind a metal can, and the sensors here will alarm if you try to cut through (you can see the little sensor lines there). So what does this actually achieve? Well, assuming that the tamper-protection works (and it appears to), the software most people run on it – IBM's Common Cryptographic Architecture (CCA) – introduces a quite general concept of typing, namely control vectors.

A control vector can be thought of as a string bound to each key. The physical implementation of this is that you encipher the key under a different variant of a master key, but that's irrelevant at the level of abstraction that's useful to think about here. You can think of each key as going into the machine with a little tag attached to it saying, "I am a green key", "I am a red key", "I am a key of type 3126", or whatever. By default, CCA provides ten different types: there is a data key; there's a data translation key; there's a MAC key; there's a PIN generation key; there's an incoming PIN encryption key; and so on. There are somewhat over 100 different transactions supplied with CCA that operate using these key types. In addition, you can define your own key types and roll your own transactions. You can download the details from the Web, all 400 and whatever pages of it [7], so the target is public domain.

Even if you use the standard default types, there are some very obscure issues that are left to the designer. There are some possible hacks, or at least unclear

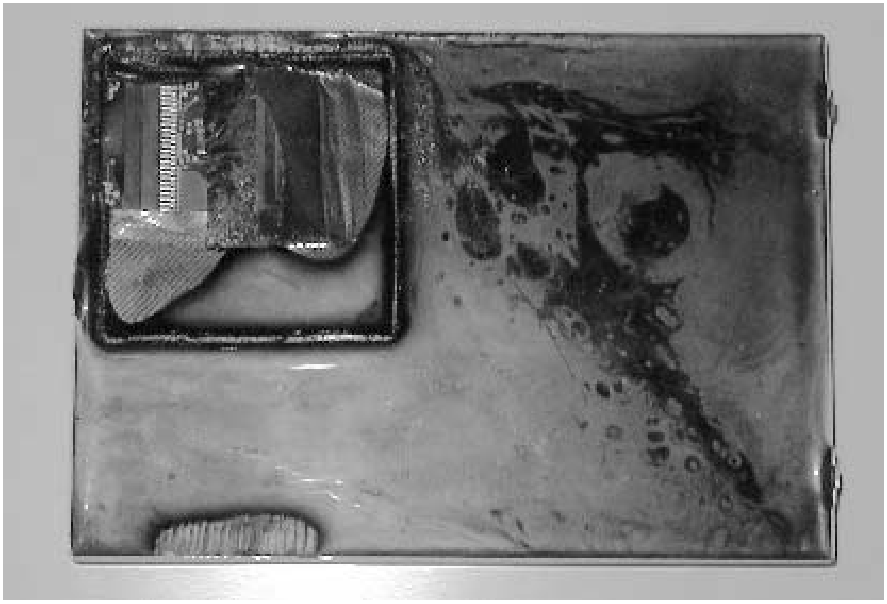

Fig. 2. – The 4758 partially opened – showing (from top left downwards) the circuitry, aluminium electromagnetic shielding, tamper sensing mesh and potting material (courtesy of Frank Stajano)

things, such as if people decrypt rubbish and use it, or if people use the wrong type of key, then can you gain some kind of advantage? Well keys supposedly have parity, so you have to decrypt on average about 32,000 'wrong' keys before you get one that works, but there have been attacks in the past which have used this [6]. Do any of these attacks work on the 4758?

Also, in the typing structure, we find that some of these types have got fairly explicit limitations (such as whether export is allowed or not), and you've got some types of types. But there's also a load of stuff that appears to be left more or less implicit, or at the very least has no supplied formal model.

You have got backward-compatibility modes. The 4758 will do everything that the old 3848 would do: it will do the ECPH, DCPH, re-format to master key, all that stuff, so you get free documentation of the obsolete equipment in the new equipment. You have things that approximate to the VISA transactions.

You also have a claim that the instruction set is comprehensive. Now this makes me feel rather uneasy, because the whole reason that I'm paying out all this money for a security processor is that its instruction set should *not* be comprehensive. There should be some things that it is simply not possible to do with it, and these things should be clearly understood.

So how did the current design come about? Well it should now be clear. We started off with the 3848, and that was not enough for banking so someone (actually Carl Campbell) invented the VISA security module. Then IBM saw itself losing market share to the VSM, so it does the embrace-and-expand play and incorporates the VSM functionality in the 4753. This evolves into the 4758, which was then used for things like prepayment electricity meters [3]. People then invented i-buttons and smartcards, yet more applications with which IBM's product line had to be compatible. Roger's phrase, "the inevitable evolution of the Swiss army knife", is very appropriate.

3 Verifying Crypto Transaction Sets

So how can you be sure that there isn't some chain of 17 transactions which will leak a clear key? I've actually got some experience of this, because with a security module that I designed there was one particular banking customer who said, we need a transaction to translate PINs. They were upgrading all of their customers to longer account numbers, and they didn't want to change the customers' PINs because they were afraid that this would decrease the number of people using their credit cards in cash machines. So they said, "We'd like to supply two account numbers, an old account number and a new account number, plus a PIN, and you return the difference between the old PIN and the new PIN so we can write that as an offset onto the card track."

So I said, "Fine – we'll code that up. But be warned: this transaction is dangerous. Remove this version of the software as soon as you've done that batch job." Of course they didn't, and about a year and a half later one of their programmers realised: "Hey – if I put in my account number and the MD's account number and my PIN, the machine then tells me what I've got to subtract from my PIN in order to get the MD's PIN!" Luckily for the bank, he went to the boss and told all; to which our response was, "We told you, why didn't you do as you were told?" (This episode is described in [5].)

So managing this kind of thing isn't as straightforward as you might think. As I mentioned, we have got no proper formal model of what goes on in a device like this. Although it has some flavours of Bell-LaPadula and some flavours of Clark-Wilson, it doesn't really behave according to either model; there are leaks at the edges. This brings us to all the research opportunities, which aren't necessarily 'insurmountable' but are, I suspect, a step change more difficult than the attacks on protocols that people have being doing up to now.

The first opportunity is to find an attack on the 4758. Find some property of the manual which even IBM will be forced to admit was not something they had in mind when they designed it. Alternatively prove that it's secure (though given the history of provable security, that's perhaps just as damning). Or, more practically, provide a tool that enables the designer to use the 4758 safely. Up until now, the IBM philosophy appears to have been: "We're selling you an F15 and you had jolly well better let us sell you a pilot and a maintenance crew and all the rest of it at the same time." They are not alone in this respect; BBN and all the other vendors appear to take the same view.

So you find that you can go and buy a set of tamper-resistant hardware boxes for maybe $2,000 each, but if you are a bank and you are building an actual installation using them, such as a certification authority, the outlay at the end of the day is more likely to be a quarter of a million dollars. By the time you've paid for the all the consultancy, and the machines they security modules will be attached to, and the firewalls to protect these machines from the network, and the procedures, and the training, and all the rest of it, the cost of the tamper-resistant box itself is essentially trivial.

So if anybody else is interested in getting into the systems integration business and getting that $240,000 instead of funneling it to IBM, then the competitive advantage might consist in having a suitable tool. Now it's not just the 4758! I use that as the target because we don't know how to break its physical tamper resistance, but there are many other architectures where you get a defined set of crypto transactions that may in fact be quite extensible. Those that come to mind are Microsoft CAPI, Intel CDSA – there's dozens out there.

This is definitely more difficult than doing verification of crypto protocols – and more interesting I think – because this is the real world problem. Somebody who has got a penetration of your corporate network is not trying to attack Kerberos, he's trying to attack the boxes that contain the Kerberos keys. Those boxes will do 50 transactions, of which you might normally use only three or four. But the others are there and available, unless you find some way of switching them off in a particular application, which again brings together a whole new set of design issues. So what's the nature of a crypto transaction set in general, and how do you go about verifying it?

Next question: how do you relate its properties to a higher level security policy model, such as Bell-LaPadula? And where this I suppose leads to, from the theory and semantics point of view, is whether you can get a general theory of restricted computation of computers that can do some things but cannot – absolutely, definitely, provably, cannot – do other things. Is there such a language that is complete in the sense that you can express any worthwhile crypto transaction set in it? What do you have to add to mathematics in order to get the expressiveness of metal? In the RIP-stop application it may very well be that provable destruction of a key is all you need, and anything else can be built on top of that given various external services such as perhaps secure time. But provable destruction of a key doesn't seem obviously related to what's going on in banking, because a typical bank is still running with the same ATM keys that they set up in 1985 when VISA set up the ACDS network. Key expiration just doesn't come into their universe.

And finally, if you're going to develop such a language, it would be useful if it could deal with public key stuff as well, such as homomorphic attacks, blinding, and so on. Because if you're going to build a machine which will, for example, not decrypt the same PGP message twice, then you have got somehow to make sure that the bad guys don't send you a message which is an old message which has been blinded so it looks like a new message.

Matt Blaze: I was surprised by your early comment – which I realise was a throwaway – that tamper-resistant hardware appears to be almost done, so let's move on to the next thing. It seems to me that tamper-resistant hardware if anything has become increasingly tenuous of late, particularly with respect to the use of out-of-band attacks such as power analysis and timing attacks. Things have been made even more difficult by putting the power supply outside of the box. Do you think once Paul Kocher gets his hands on one of these IBM boxes and hooks up his spectrum analyser to it that that's the end?

Reply: In the case of the IBM product I don't think so, because they've got a reasonably convincing proof (in the electrical engineering sense) that this can't happen. They filter the power lines into the device in such a way that all the interesting signals, given the frequencies that they're at inside the box, are attenuated more than 100 dBs. I agree that there is a serious problem with smartcards, but we've got a project going on that, we believe it to be fixable. We discuss this in a paper at a VLSI conference recently [9]. What we're proposing to do is to use self-timed dual-rail logic, so that the current that is consumed by doing a computation is independent of the data that is being operated on. You can do various other things as well; you can see to it that no single transistor failure will result in an output, and so on. In fact the NSA already uses dual-rail for resilience rather than power-analysis reasons. So there are technologies coming along that can fix the power analysis problem.

When I say that hardware tamper-resistance has been done, what I'm actually saying is that we can see our way to products on the shelf in five or seven years time that will fix the kind of things that Paul Kocher has been talking about.

John Ioannidis: The kind of tamper-resistance you seem to analyse here is not the sort of tamper resistance I'm likely to find in my wallet or my key-ring. It's more like the sort of tamper-resistance you find in a big box somewhere. Maybe that big box is sitting on my desk, or under my desk, but it's not going to be sitting on my person.

Reply: Wrong. The logical complexity that I've described here is independent of the physical size of the device. You consider the kind of SIM card that's likely to appear, with added-on transactions to support an electronic purse, and you're already looking at the same scale of complexity as a VISA security module. You're looking at much more difficult management problems, because you have got independent threads of trust: the phone company trusts its K_c and the bank trusts its EMV keys, and how do these devices interact within the smartcard itself? It suddenly becomes very hairy and we don't at present have the tools to deal with it.

Virgil Gligor: How many 4758s did IBM sell, do you know what proportion were penetrated relative to NT or Windows 98, or any of the things that you have on your desk?

Reply: They're selling of the order of a couple of thousand a year, I'm aware of only one penetration which occurred – during testing as a result of a manufacturing defect.

I assume that an NT box, or for that matter a Linux box, can be penetrated by somebody who knows what he's doing, and so you must assume that the host to which your security module is attached belongs to the enemy, even if only from time to time.

(Indistinct question)

Reply: It's useful in security to separate design assurance from implementation assurance and from strength of mechanism. Now the strength of mechanism of the tamper-resistance mechanisms in 4758s are very much higher than a smartcard, and a smartcard is very much higher than an NT server, but that's a separate lecture course.

What I'm concerned about here is the logical complexity of the set of commands which you issue to your tamper resistant device, be it a smartcard, or an NT server, or a 4758, because if you screwed up somehow in the design of those, then it doesn't matter how much steel and how many capacitors and how many seismometers and how many men with guns you've got. You're dead.

(Indistinct question)

Reply: We brought the designer of the 4758 over and we grilled him for a day, and we've looked at the things partially dismantled, and we've got one or two off-the-wall ideas about how you might conceivably, if you spent a million bucks building equipment, have a go at it. But we don't have any practical ideas that we could offer to a corporation and say, "Give us a million bucks and with a better than 50% chance we'll have the keys out in a year." The 4758 is hard!

John Ioannidis: What kind of primitives am I going to be offloading on to tamper-proof hardware so that no matter what kind of penetration my NT box has, there won't be an order for a billion dollars to be transferred from somewhere else to my account without my consent, so I can't get framed for fraud.

Reply: The sort of thing that hardware boxes are traditionally used for is to protect the PIN key, the key that generates all the PINs of the eleven million customers in your bank, because if a bad guy gets hold of that you've had it, you have to re-issue your cards and PINs. The sort of thing that you can use a 4758 for in a high-value corporate environment is to see to it that a transaction will only go out signed with the great seal of Barclays Bank (or whatever) provided – let's say – in the case of a $100m transaction, at least two managers of grade 4 or above have signed it using the signing keys and their smartcards. You can set up a system involving the 4758 and the smartcards which will distribute the keys, manage the certificates, and then will basically enforce your Clark-Wilson policy.

John Ioannidis: I'm not a grade 4 manager at Goldman Sachs, or whatever they're called, Suppose I'm a poor AT&T employee, and I have a box on my desk and I do my banking transactions over it, and I have somewhere a key given me by the bank, or I created my own, or both, and I have a tamper-resistant device which I want to use to secure my transaction. Now the question is, what kind of operations should that thing support, and what kind of interface to me, that does not go through NT or Linux, so that a transaction could not be made without my personal, physical approval – not my electronic approval. This is also a hard problem.

Reply: Agreed. But it's not the problem I'm addressing. Many people have talked about signature tablets that will display text; you put your thumb print, for example, to OK this text and this is supposed to give you a high degree of non-repudiation. But that's a different problem; it's not the problem that interests me in the context of this talk.

Michael Roe: What Ross is deliberately ignoring is the effect of successful tamper-resistance. If you have a very secure transaction in a very physically secure processor then the attacker is going to attack the software that goes in between the user interface and the secure box.

Reply: In this talk what I'm interested in is the command set that you use to drive a tamper resistant device and how you go about defining that in a way that's sensible – and assuring yourself that the design holds up. (I'm talking about design assurance not about the human computer interface aspects.)

(Indistinct question)

Reply: There have been attacks on early pay TV systems where there were protocol failures in the smartcard. Now if you believe Gemplus figures that by 2003 there will be 1.4 billion microprocessor cards sold a year (and as the biggest OEM they should know about that) then clearly it is important in engineering and industrial and economic terms, as well as being a problem that's directly related to protocol verification. That's the reason why I bring it to this audience. It is a protocol problem; but it's the next step up in difficulty. And it's sufficiently different that there's an awful lot of new research for people to do, and, hopefully, new theories to discover.

(Indistinct question)

Reply: I'm interested in Clark-Wilson-type objects and Bell-LaPadula-type objects, because that's where the real world is. The usual objection to Clark-Wilson is, "Where's the TCB?" Now as an example application, banking security modules are 90% of the way towards a Clark-Wilson TCB, so this should also be of scientific interest.

(Indistinct question)

Reply: An awful lot of design effort is believed to be about to go into multifunction smartcards, where you've got a SIM card sitting in your WAP device, and into which signed applets can be downloaded by your bank, by your insurance company, by your health care provider, etc.

(Indistinct question)

Reply: Now separability is not the only issue here, there's the correctness of the transactions that people can issue, and whether the applications themselves can be sabotaged by somebody just issuing the right commands in order.

(Indistinct question)

Reply: Even if you can download these applets safely, the applets will have to talk to each other in many cases for it to be useful. For example, if you're going to use your WAP device to pay for hospital treatment, by transferring money from your electronic purse to your health card, then that involves having interfaces that allow transactions between the two, and that's hard to do.

References

1. RJ Anderson, *'Security Engineering – a Guide to Building Dependable Distributed Systems'*, Wiley (2001) ISBN 0-471-38922-6
2. RJ Anderson, "Why Cryptosystems Fail" in *Communications of the ACM* vol 37 no 11 (November 1994) pp 32–40; earlier version at `http://www.cl.cam.ac.uk/users/rja14/wcf.html`
3. RJ Anderson, SJ Bezuidenhoudt, "On the Reliability of Electronic Payment Systems", in *IEEE Transactions on Software Engineering* vol 22 no 5 (May 1996) pp 294–301; `http://www.cl.cam.ac.uk/ftp/users/rja14/meters.ps.gz`
4. RJ Anderson, MG Kuhn, "Tamper Resistance – a Cautionary Note", in *Proceedings of the Second Usenix Workshop on Electronic Commerce* (Nov 96) pp 1–11; `http://www.cl.cam.ac.uk/users/rja14/tamper.html`
5. RJ Anderson, MG Kuhn, "Low Cost Attacks on Tamper Resistant Devices", in *Security Protocols – Proceedings of the 5th International Workshop* (1997) Springer LNCS vol 1361 pp 125–136
6. M Blaze, "Protocol Failure in the Escrowed Encryption Standard", in *Second ACM Conference on Computer and Communications Security*, 2–4 November 1994, Fairfax, Va; proceedings published by the ACM ISBN 0-89791-732-4, pp 59–67; at `http://www.crypto.com/papers/`
7. IBM, *'IBM 4758 PCI Cryptographic Coprocessor – CCA Basic Services Reference and Guide*, Release 1.31 for the IBM 4758-001, available through `http://www.ibm.com/security/cryptocards/`
8. CH Meyer, SM Matyas, *'Cryptography: A New Dimension in Computer Data Security'*, Wiley, 1982
9. SW Moore, RJ Anderson, MG Kuhn, "Improving Smartcard Security using Self-timed Circuit Technology", Fourth ACiD-WG Workshop, Grenoble, ISBN 2-913329-44-6, 2000; at `http://www.g3card.com/`
10. GJ Simmons, "How to Insure that Data Acquired to Verify Treaty Compliance are Trustworthy", GJ Simmons, *Proceedings of the IEEE* v 76 no 5 (1988)

Why Cryptosystems Fail

Ross Anderson
University Computer Laboratory
Pembroke Street, Cambridge CB2 3QG
Email: rja14@cl.cam.ac.uk

This paper is also available in pdf and postscript.

Abstract:

Designers of cryptographic systems are at a disadvantage to most other engineers, in that information on how their systems fail is hard to get: their major users have traditionally been government agencies, which are very secretive about their mistakes.

In this article, we present the results of a survey of the failure modes of retail banking systems, which constitute the next largest application of cryptology. It turns out that the threat model commonly used by cryptosystem designers was wrong: most frauds were not caused by cryptanalysis or other technical attacks, but by implementation errors and management failures. This suggests that a paradigm shift is overdue in computer security; we look at some of the alternatives, and see some signs that this shift may be getting under way.

Introduction

Cryptology, the science of code and cipher systems, is used by governments, banks and other organisations to keep information secure. It is a complex subject, and its national security overtones may invest it with a certain amount of glamour, but we should never forget that information security is at heart an engineering problem. The hardware and software products which are designed to solve it should in principle be judged in the same way as any other products: by their cost and effectiveness.

However, the practice of cryptology differs from, say, that of aeronautical engineering in a rather striking way: there is almost no public feedback about how cryptographic systems fail.

When an aircraft crashes, it is front page news. Teams of investigators rush to the scene, and the subsequent enquiries are conducted by experts from organisations with a wide range of interests - the carrier, the insurer, the manufacturer, the airline pilots' union, and the local aviation authority. Their findings are examined by journalists and politicians, discussed in pilots' messes, and passed on by flying instructors.

In short, the flying community has a strong and institutionalised learning mechanism. This is perhaps the main reason why, despite the inherent hazards of flying in large aircraft, which are maintained and piloted by fallible human beings, at hundreds of miles an hour through congested airspace, in bad weather and at night, the risk of being killed on an air journey is only about one in a million.

In the crypto community, on the other hand, there is no such learning mechanism. The history of the subject ([K1], [W1]) shows the same mistakes being made over and over again; in particular, poor management of codebooks and cipher machine procedures enabled many communication networks to be broken. Kahn relates, for example [K1, p 484], that Norway's rapid fall in the second world war was largely due to the fact that the British Royal Navy's codes had been solved by the German Beobachtungsdienst - using exactly the same techniques that the Royal Navy's own `Room 40' had used against Germany in the previous war.

Since world war two, a curtain of silence has descended on government use of cryptography. This is not surprising, given not just the cold war, but also the reluctance of bureaucrats (in whatever organisation) to admit their failures. But it does put the cryptosystem designer at a severe disadvantage compared with engineers working in other disciplines; the post-war years are precisely the period in which modern cryptographic systems have been developed and brought into use. It is as if accident reports were only published for piston-engined aircraft, and the causes of all jet aircraft crashes were kept a state secret.

Automatic Teller Machines

To discover out how modern cryptosystems are vulnerable in practice, we have to study their use elsewhere. After government, the next biggest application is in banking, and evolved to protect automatic teller machines (ATMs) from fraud.

In some countries (including the USA), the banks have to carry the risks associated with new technology. Following a legal precedent, in which a bank customer's word that she had not made a withdrawal was found to outweigh the banks' experts' word that she must have done [JC], the US Federal Reserve passed regulations which require banks to refund all disputed transactions unless they can prove fraud by the customer [E]. This has led to some minor abuse - misrepresentations by customers are estimated to cost the average US bank about $15,000 a year [W2] - but it has helped promote the development of security technologies such as cryptology and video.

In Britain, the regulators and courts have not yet been so demanding, and despite a parliamentary commission of enquiry which found that the PIN system was insecure [J1], bankers simply deny that their systems are ever at fault. Customers who complain about debits on their accounts for which they were not responsible - so-called `phantom withdrawals' - are told that they are lying, or mistaken, or that they must have been defrauded by their friends or relatives.

The most visible result in the UK has been a string of court cases, both civil and criminal. The pattern which emerges leads us to suspect that there may have been a number of miscarriages of justice over the years.

- A teenage girl in Ashton under Lyme was convicted in 1985 of stealing £40 from her father. She pleaded guilty on the advice of her lawyers that she had no defence, and then disappeared; it later turned out that there had been never been a theft, but merely a clerical error by the bank [MBW]
- A Sheffield police sergeant was charged with theft in November 1988 and suspended for almost a year after a phantom withdrawal took place on a card he had confiscated from a suspect. He was lucky in that his colleagues tracked down the lady who had made the transaction after the disputed one; her eyewitness testimony cleared him
- Charges of theft against an elderly lady in Plymouth were dropped after our enquiries showed that the bank's computer security systems were a shambles
- In East Anglia alone, we are currently advising lawyers in two cases where people are awaiting trial for alleged thefts, and where the circumstances give reason to believe that `phantom withdrawals' were actually to blame.

Finally, in 1992, a large class action got underway in the High Court in London [MB], in which hundreds of plaintiffs seek to recover damages from various banks and building societies. We were retained by the plaintiffs to provide expert advice, and accordingly conducted some research during 1992 into the actual and possible failure modes of automatic teller machine systems. This involved interviewing former bank employees and criminals, analysing statements from plaintiffs and other victims of ATM fraud, and searching the literature. We were also able to draw on experience gained during the mid-80's on designing cryptographic equipment for the financial sector, and advising clients overseas on its use.

We shall now examine some of the ways in which ATM systems have actually been defrauded. We will then compare them with how the designers thought their products might in theory be vulnerable, and see what lessons can be drawn. Some material has had to be held back for legal reasons, and in particular we do not identify all the banks whose mistakes we discuss. This information should be provided by witnesses at trial, and its absence here should have no effect on the points we wish to make.

How ATM Fraud Takes Place

We will start with some simple examples which indicate the variety of frauds that can be carried out without any great technical sophistication, and the bank operating procedures which let them happen. For the time being, we may consider that the magnetic strip on the customer's card contains only his account number, and that his personal identification number (PIN) is derived by encrypting this account number and taking four digits from the result. Thus the ATM must be able to perform this encryption operation, or to check the PIN in some other way (such as by an online enquiry).

Some simple examples

1.

Many frauds are carried out with some inside knowledge or access, and ATM fraud turns out to be no exception. Banks in the English speaking world dismiss about one percent of their staff every year for disciplinary reasons, and many of these sackings are for petty thefts in which ATMs can easily be involved. A bank with 50,000 staff, which issued cards and PINs through the branches rather than by post, might expect about two incidents per business day of staff stealing cards and PINs.

- In a recent case, a housewife from Hastings, England, had money stolen from her account by a bank clerk who issued an extra card for it. The bank's systems not only failed to prevent this, but also had the feature that whenever a cardholder got a statement from an ATM, the items on it would not subsequently appear on the full statements sent to the account address. This enabled the clerk to see to it that she did not get any statement showing the thefts he had made from her account.

  This was one of the reasons he managed to make 43 withdrawals of £200 each; the other was that when she did at last complain, she was not believed. In fact she was subjected to harrassment by the bank, and the thief was only discovered because he suffered an attack of conscience and owned up [RM].

- Technical staff also steal clients' money, knowing that complaints will probably be ignored. At one bank in Scotland, a maintenance engineer fitted an ATM with a handheld computer, which recorded customers' PINs and account numbers. He then made up counterfeit cards and looted their accounts [C1] [C2]. Again, customers who complained were stonewalled; and the bank was publicly criticised for this by one of Scotland's top law officers.

- One bank issues tellers with cards with which they can withdraw money from branch ATMs and debit any customer account. This may be convenient when the teller station cash runs out, but could lead the staff into temptation.

- One bank had a well managed system, in which the information systems, electronic banking and internal audit departments cooperated to enforce tight dual control over unissued cards and PINs in the branches. This kept annual theft losses down, until one day a protegé of the deputy managing director sent a circular to all branches announcing that to cut costs, a number of dual control procedures were being abolished, including that on cards and PINs. This was done without consultation, and without taking any steps to actually save money by reducing staff. Losses increased tenfold; but managers in the affected departments were unwilling to risk their careers by making a fuss. This seems to be a typical example of how computer security breaks down in real organisations.

Most thefts by staff show up as phantom withdrawals at ATMs in the victim's neighbourhood. English banks maintain that a computer security problem would result in a random distribution of transactions round the country, and as most disputed withdrawals happen near the customer's home or place of work, these must be due to cardholder negligence [BB]. Thus the pattern of complaints which arises from thefts by their own staff only tends to reinforce the banks' complacency about their systems.

2. Outsiders have also enjoyed some success at attacking ATM systems.

   - In a recent case at Winchester Crown Court in England [RSH], two men were convicted of a simple but effective scam. They would stand in ATM queues, observe customers' PINs, pick up the discarded ATM tickets, copy the account numbers from the tickets to blank cards, and use these to loot the customers' accounts.

     This trick had been used (and reported) several years previously at a bank in New York. There the culprit was an ATM technician, who had been fired, and who managed to steal over $80,000 before the bank saturated the area with security men and caught him in the act.

     These attacks worked because the banks printed the full account number on the ATM ticket, and because there was no cryptographic redundancy on the magnetic strip. One might have thought that the New York lesson would have been learned, but no: in England, the bank which had been the main victim in the Winchester case only stopped printing the full account number in mid 1992, after the author replicated the fraud on television to warn the public of the risk. Another bank continued printing it into 1993, and was pilloried by journalists who managed to forge a card and use it [L1].

   - Another technical attack relies on the fact that most ATM networks do not encrypt or authenticate the authorisation response to the ATM. This means that an attacker can record a `pay' response from the bank to the machine, and then keep on replaying it until the machine is empty. This technique, known as `jackpotting', is not limited to outsiders - it appears to have been used in 1987 by a bank's operations staff, who used network control devices to jackpot ATMs where accomplices were waiting.

   - Another bank's systems had the feature that when a telephone card was entered at an ATM, it believed that the previous card had been inserted again. Crooks stood in line, observed customers' PINs, and helped themselves. This shows how even the most obscure programming error can lead to serious problems.

   - Postal interception is reckoned to account for 30% of all UK payment card losses [A1], but most banks' postal control procedures are dismal. For example, in February 1992 the author asked for an increased card limit: the bank sent not one, but two, cards and PINs through the post. These cards arrived only a few days after intruders had got hold of our apartment block's mail and torn it up looking for valuables.

It turned out that this bank did not have the systems to deliver a card by registered post, or to send it to a branch for collection. Surely they should have noticed that many of their Cambridge customers live in colleges, student residences and apartment buildings which have no secure postal deliveries; and that most of the new students open bank accounts at the start of the academic year in October, when large numbers of cards and PIN mailers are left lying around on staircases and in pigeonholes.

- Test transactions have been another source of trouble. There was a feature on one make of ATM which would output ten banknotes when a 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. These went on until all the banks using the machine put in a software patch to disable the transaction.

- The fastest growing modus operandi is to use false terminals to collect customer card and PIN data. Attacks of this kind were first reported from the USA in 1988; there, crooks built a vending machine which would accept any card and PIN, and dispense a packet of cigarettes. They put their invention in a shopping mall, and harvested PINs and magnetic strip data by modem. A more recent instance of this in Connecticut got substantial press publicity [J2], and the trick has spread to other countries too: in 1992, criminals set up a market stall in High Wycombe, England, and customers who wished to pay for goods by credit card were asked to swipe the card and enter the PIN at a terminal which was in fact hooked up to a PC. At the time of writing, British banks had still not warned their customers of this threat.

3.

The point of using a four-digit PIN is that someone who finds or steals another person's ATM card has a chance of only one in ten thousand of guessing the PIN, and if only three attempts are allowed, then the likelihood of a stolen card being misused should be less than one in 3,000. However, some banks have managed to reduce the diversity of a four-digit PIN to much less than 10,000. For example:

- They may have a scheme which enables PINs to be checked by offline ATMs and point-of-sale devices without these devices having a full encryption capability. For example, customers of one bank get a credit card PIN with digit one plus digit four equal to digit two plus digit three, and a debit card PIN with one plus three equals two plus four. This means that crooks could use stolen cards in offline devices by entering a PIN such as 4455.

- In early 1992, another bank sent its cardholders a letter warning them of the dangers of writing their PIN on their card, and suggested instead that they conceal the PIN in the following way and write it down on a distinctive piece of squared cardboard, which was designed to be kept alongside the ATM card in a wallet or purse.

Suppose your PIN is 2256. Choose a four-letter word, say `blue'. Write these four letters down in the second, second, fifth and sixth columns of the card respectively:

1	2	3	4	5	6	7	8	9	0
	b								
	l								
				u					
					e				

Now fill up the empty boxes with random letters. Easy, isn't it? Of course, there may be only about two dozen four-letter words which can be made up using a given grid of random letters, so a thief's chance of being able to use a stolen card has just increased from 1 in 3,333 to 1 in 8.

- One small institution issued the same PIN to all its customers, as a result of a simple programming error. In yet another, a programmer arranged things so that only three different PINs were issued, with a view to forging cards by the thousand. In neither case was the problem detected until some considerable time had passed: as the live PIN mailers were subjected to strict handling precautions, no member of staff ever got hold of more than his own personal account mailer.

4. Some banks do not derive the PIN from the account number by encryption, but rather chose random PINs (or let the customers choose them) and then encrypt them for storage. Quite apart from the risk that customers may choose PINs which are easy to guess, this has a number of technical pitfalls.

   - Some banks hold the encrypted PINs on a file. This means that a programmer might observe that the encrypted version of his own PIN is (say) 132AD6409BCA4331, and search the database for all other accounts with the same PIN.

- One large UK bank even wrote the encrypted PIN to the card strip. It took the criminal fraternity fifteen years to figure out that you could change the account number on your own card's magnetic strip to that of your target, and then use it with your own PIN to loot his account.

  In fact, the Winchester pair used this technique as well, and one of them wrote a document about it which appears to have circulated in the UK prison system [S]; and there are currently two other men awaiting trial for conspiring to defraud this bank by forging cards.

For this reason, VISA recommends that banks should combine the customer's account number with the PIN before encryption [VSM]. Not all of them do.

5\.

Despite all these horrors, Britain is by no means the country worst affected by card forgery. That dubious honour goes to Italy [L2], where losses amount to almost 0.5% of ATM turnover. Banks there are basically suffering from two problems.

- The first is a plague of bogus ATMs - devices which look like real ATMs, and may even be real ATMs, but which are programmed to capture customers' card and PIN data. As we saw above, this is nothing new and should have been expected.

- The second is that Italy's ATMs are generally offline. This means that anyone can open an account, get a card and PIN, make several dozen copies of the card, and get accomplices to draw cash from a number of different ATMs at the same time. This is also nothing new; it was a favourite modus operandi in Britain in the early 1980's [W3].

More complex attacks

The frauds which we have described so far have all been due to fairly simple errors of implementation and operation. Security researchers have tended to consider such blunders uninteresting, and have therefore concentrated on attacks which exploit more subtle technical weaknesses. Banking systems have a number of these weaknesses too.

Although high-tech attacks on banking systems are rare, they are of interest from the public policy point of view, as government initiatives such as the EC's Information Technology Security Evaluation Criteria [ITSEC] aim to develop a pool of evaluated products which have been certified free of known technical loopholes.

The basic assumptions behind this program are that implementation and operation will be essentially error-free, and that attackers will possess the technical skills which are available in a government signals security agency. It would therefore seem to be more relevant to military than civilian systems, although we will have more to say on this later.

In order to understand how these sophisticated attacks might work, we must look at banking security systems in a little more detail.

How ATM encryption works

Most ATMs operate using some variant of a system developed by IBM, which is documented in [MM]. This uses a secret key, called the `PIN key', to derive the PIN from the account number, by means of a published algorithm known as the Data Encryption Standard, or DES. The result of this operation is called the `natural PIN'; an offset can be added to it in order to give the PIN which the customer must enter. The offset has no real cryptographic function; it just enables customers to choose their own PIN. Here is an example of the process:

Account number:	8807012345691715
PIN key:	FEFEFEFEFEFEFEFE
Result of DES:	A2CE126C69AEC82D
Result decimalised:	0224126269042823
Natural PIN:	0224
Offset:	6565
Customer PIN:	6789

It is clear that the security of the system depends on keeping the PIN key absolutely secret. The usual strategy is to supply a `terminal key' to each ATM in the form of two printed components, which are carried to the branch by two separate officials, input at the ATM keyboard, and combined to form the key. The PIN key, encrypted under this terminal key, is then sent to the ATM by the bank's central computer.

If the bank joins a network, so that customers of other banks can use its ATMs, then the picture becomes more complex still. `Foreign' PINs must be encrypted at the ATM using a `working' key it shares with its own bank, where they are decrypted and immediately re-encrypted using another working key shared with the card issuing bank.

These working keys in turn have to be protected, and the usual arrangement is that a bank will share a `zone key' with other banks or with a network switch, and use this to encrypt fresh working keys which are set up each morning. It may also send a fresh working key every day to each of its ATMs, by encrypting it under the ATM's terminal key.

A much fuller description of banking security systems can be found in books such as [DP] and [MM], and in equipment manuals such as [VSM] and [NSM]. All we really need to know is that a bank has a number of keys which it must keep secret. The most important of these is of course the PIN key, as anyone who gets hold of this can forge a card for any customer's account; but other keys (such as terminal keys, zone keys and working keys) could also be used, together with a wiretap, to find out customer PINs in large numbers.

Keeping keys secret is only part of the problem. They must also be available for use at all times by authorised processes. The PIN key is needed all the time to verify transactions, as are the current working keys; the terminal keys and zone keys are less critical, but are still used once a day to set up new working keys.

The original IBM encryption products, such as PCF and the 3848, did not solve the problem: they only did the encryption step, and left the other manipulations to a mainframe computer program, which each bank had to write anew for itself. Thus the security depended on the skill and integrity of each bank's system development and maintenance staff.

The standard approach nowadays is to use a device called a security module. This is basically a PC in a safe, and it is programmed to manage all the bank's keys and PINs in such a way that the mainframe programmers only ever see a key or PIN in encrypted form. Banks which belong to the VISA and Mastercard ATM networks are supposed to use security modules, in order to prevent any bank customer's PIN becoming known to a programmer working for another bank (the Mastercard security requirements are quoted in [MM]; for VISA see [VSM]).

Problems with encryption products

In practice, there are a number of problems with encryption products, whether the old 3848s or the security modules now recommended by banking organisations. No full list of these problems, whether actual or potential, appears to have been published anywhere, but they include at least the following which have come to our notice:

1. Although VISA and Mastercard have about 10,000 member banks in the USA and at least 1,000 of these do their own processing, enquiries to security module salesmen reveal that only 300 of these processing centres had actually bought and installed these devices by late 1990. The first problem is thus that the hardware version of the product does not get bought at all, either because it is felt to be too expensive, or because it seems to be too difficult and time-consuming to install, or because it was not supplied by IBM (whose own security module product, the 4753, only

became available in 1990). Where a bank has no security modules, the PIN encryption functions will typically be performed in software, with a number of undesirable consequences.

- The first, and obvious, problem with software PIN encryption is that the PIN key can be found without too much effort by system programmers. In IBM's product, PCF, the manual even tells how to do this. Once armed with the PIN key, programmers can easily forge cards; and even if the bank installs security modules later, the PIN key is so useful for debugging the systems which support ATM networking that knowledge of it is likely to persist among the programming staff for years afterward.

- Programmers at one bank did not even go to the trouble of setting up master keys for its encryption software. They just directed the key pointers to an area of low memory which is always zero at system startup. The effect of this was that the live and test systems could use the same cryptographic key dataset, and the bank's technicians found that they could work out customer PINs on their test equipment. Some of them used to charge the local underworld to calculate PINs on stolen cards; when the bank's security manager found that this was going on, he was killed in a road accident (of which the local police conveniently lost the records). The bank has not bothered to send out new cards to its customers.

2.

The `buy-IBM-or-else' policy of many banks has backfired in more subtle ways. One bank had a policy that only IBM 3178 terminals could be purchased, but the VISA security modules they used could not talk to these devices (they needed DEC VT 100s instead). When the bank wished to establish a zone key with VISA using their security module, they found they had no terminal which would drive it. A contractor obligingly lent them a laptop PC, together with software which emulated a VT100. With this the various internal auditors, senior managers and other bank dignitaries duly created the required zone keys and posted them off to VISA.

However, none of them realised that most PC terminal emulation software packages can be set to log all the transactions passing through, and this is precisely what the contractor did. He captured the clear zone key as it was created, and later used it to decrypt the bank's PIN key. Fortunately for them (and VISA), he did this only for fun and did not plunder their network (or so he claims).

3.

Not all security products are equally good, and very few banks have the expertise to tell the good ones from the mediocre.

- The security module's software may have trapdoors left for the convenience of the vendor's engineers. We only found this out because one bank had no proper ATM test environment; when it decided to join a network, the vendor's systems engineer could not get the gateway working, and, out of frustration, he used one of these tricks to extract the PIN key from the system, in the hope that this would help him find the problem. The existence of such trapdoors makes it impossible to devise effective control procedures over security modules, and we have so far been lucky that none of these engineers have tried to get into the card forgery business (or been forced to cooperate with organised crime).

- Some brands of security module make particular attacks easier. Working keys may, for example, be generated by encrypting a time-of-day clock and thus have only 20 bits of diversity rather than the expected 56. Thus, according to probability theory, it is likely that once about 1,000 keys have been generated, there will be two of them which are the same. This makes possible a number of subtle attacks in which the enemy manipulates the bank's data communications so that transactions generated by one terminal seem to be coming from another.

- A security module's basic purpose is to prevent programmers, and staff with access to the computer room, from getting hold of the bank's cryptographic keys. However, the `secure' enclosure in which the module's electronics is packaged can often be penetrated by cutting or drilling. The author has even helped a bank to do this, when it lost the physical key for its security modules.

- A common make of security module implements the tamper-protection by means of wires which lead to the switches. It would be trivial for a maintenance engineer to cut these, and then next time he visited that bank he would be able to extract clear keys.

- Security modules have their own master keys for internal use, and these keys have to backed up somewhere. The backup is often in an easily readable form, such as PROM chips, and these may need to be read from time to time, such as when transferring control over a set of zone and terminal keys from one make of security module to another. In such cases, the bank is competely at the mercy of the experts carrying out the operation.

- ATM design is also at issue here. Some older makes put the encryption in the wrong place - in the controller rather than in the dispenser itself. The controller was intended to sit next to the dispenser inside a branch, but many ATMs are no longer anywhere near a bank building. One UK university had a machine on campus which sent clear PINs and account data down a phone line to a controller in its mother branch, which is several miles away in town. Anyone who borrowed a datascope and used it on this line could have forged cards by the thousand.

4.

Even where one of the better products is purchased, there are many ways in which a poor implementation or sloppy operating procedures can leave the bank exposed.

- Most security modules return a whole range of response codes to incoming transactions. A number of these, such as `key parity error' [VSM] give advance warning that a programmer is experimenting with a live module. However, few banks bother to write the device driver software needed to intercept and act on these warnings.

- We know of cases where a bank subcontracted all or part of its ATM system to a `facilities management' firm, and gave this firm its PIN key. There have also been cases where PIN keys have been shared between two or more banks. Even if all bank staff could be trusted, outside firms may not share the banks' security culture: their staff are not always vetted, are not tied down for life with cheap mortgages, and are more likely to have the combination of youth, low pay, curiosity and recklessness which can lead to a novel fraud being conceived and carried out.

- Key management is usually poor. We have experience of a maintenance engineer being given both of the PROMs in which the security module master keys are stored. Although dual control procedures existed in theory, the staff had turned over since the PROMs were last used, and so no-one had any idea what to do. The engineer could not only have forged cards; he could have walked off with the PROMs and shut down all the bank's ATM operations.

- At branch level, too, key management is a problem. As we have seen, the theory is that two bankers type in one key component each, and these are combined to give a terminal master key; the PIN key, encrypted under this terminal master key, is then sent to the ATM during the first service transaction after maintenance.

  If the maintenance engineer can get hold of both the key components, he can decrypt the PIN key and forge cards. In practice, the branch managers who have custody of the keys are quite happy to give them to him, as they don't like standing around while the machine is serviced. Furthermore, entering a terminal key component means using a keyboard, which many older managers consider to be beneath their dignity.

- We have accounts of keys being kept in open correspondence files, rather than being locked up. This applies not just to ATM keys, but also to keys for interbank systems such as SWIFT, which handles transactions worth billions. It might be sensible to use initialisation keys, such as terminal keys and zone keys, once only and then destroy them.

- Underlying many of these control failures is poor design psychology. Bank branches (and computer centres) have to cut corners to get the day's work done, and only those control procedures whose purpose is evident are likely to be strictly observed. For example, sharing the branch safe keys between the manager and the accountant is well understood: it protects both of them from having their families taken hostage. Cryptographic keys are often not packaged in as user-friendly a way, and are thus not likely to be managed as well. Devices which actually look like keys (along the lines of military crypto ignition keys) may be part of the answer here.

- We could write at great length about improving operational procedures (this is not a threat!), but if the object of the exercise is to prevent any cryptographic key from falling into the hands of someone who is technically able to abuse it, then this should be stated as an explicit objective in the manuals and training courses. `Security by obscurity' often does more harm than good.

5. Cryptanalysis may be one of the less likely threats to banking systems, but it cannot be completely ruled out.

    - Some banks (including large and famous ones) are still using home-grown encryption algorithms of a pre-DES vintage. One switching network merely `scrambled' data blocks by adding a constant to them; this went unprotested for five years, despite the network having over forty member banks - all of whose insurance assessors, auditors and security consultants presumably read through the system specification.

    - In one case, the two defendants tried to entice a university student into helping them break a bank's proprietary algorithm. This student was studying at a maths department where teaching and research in cryptology takes place, so the skills and the reference books were indeed available. Fortunately for the bank, the student went to the police and turned them in.

    - Even where a `respectable' algorithm is used, it may be implemented with weak parameters. For example, banks have implemented RSA with key sizes between 100 and 400 bits, despite the fact that they key needs to be at least 500 bits to give any real margin of security.

    - Even with the right parameters, an algorithm can easily be implemented the wrong way. We saw above how writing the PIN to the card track is useless, unless the encryption is salted with the account number or otherwise tied to the individual card; there are many other subtle errors which can be made in designing cryptographic protocols, and the study of them is a whole discipline of itself [BAN]. In fact, there is open controversy about the design of a new banking encryption standard, ISO 11166, which is already in use by some 2,000 banks worldwide [R].

    - It is also possible to find a DES key by brute force, by trying all the possible encryption keys until you find the one which the target bank uses. The protocols used in international networks to encrypt working keys under zone keys make it easy to attack a zone key in this way: and once this has been solved, all the PINs sent or received by that bank on the network can be decrypted.

      A recent study by researchers at a Canadian bank [GO] concluded that this kind of attack would now cost about £30,000 worth of specialist computer time per zone key. It follows that it is well within the resources of organised crime, and could even be carried out by a

> reasonably well heeled individual.
>
> If, as seems likely, the necessary specialist computers have been built by the intelligence agencies of a number of countries, including countries which are now in a state of chaos, then there is also the risk that the custodians of this hardware could misuse it for private gain.

The consequences for bankers

The original goal of ATM crypto security was that no systematic fraud should be possible without the collusion of at least two bank staff [NSM]. Most banks do not seem to have achieved this goal, and the reasons have usually been implementation blunders, ramshackle administration, or both.

The technical threats described in section 3.2.2 above are the ones which most exercised the cryptographic equipment industry, and which their products were designed to prevent. However, only two of the cases in that section actually resulted in losses, and both of those can just as easily be classed as implementation failures.

The main technical lessons for bankers are that competent consultants should have been hired, and much greater emphasis should have been placed on quality control. This is urgent for its own sake: for in addition to fraud, errors also cause a significant number of disputed ATM transactions.

All systems of any size suffer from program bugs and operational blunders: banking systems are certainly no exception, as anyone who has worked in the industry will be aware. Branch accounting systems tend to be very large and complex, with many interlocking modules which have evolved over decades. Inevitably, some transactions go astray: debits may get duplicated or posted to the wrong account.

This will not be news to financial controllers of large companies, who employ staff to reconcile their bank accounts. When a stray debit appears, they demand to see a voucher for it, and get a refund from the bank when this cannot be produced. However, the ATM customer with a complaint has no such recourse; most bankers outside the USA just say that their systems are infallible.

This policy carries with it a number of legal and administrative risks. Firstly, there is the possibility that it might amount to an offence, such as conspiracy to defraud; secondly, it places an unmeetable burden of proof on the customer, which is why the US courts struck it down [JC], and courts elsewhere may follow their lead; thirdly, there is a moral hazard, in that staff are encouraged to steal by the knowledge that they are unlikely to be caught; and fourthly, there is an intelligence failure, as with no central records of customer complaints it is not possible to monitor fraud patterns properly.

The business impact of ATM losses is therefore rather hard to quantify. In the UK, the Economic Secretary to the Treasury (the minister responsible for bank regulation) claimed in June 1992 that errors affected at most two ATM transactions out of the three million which take place every day [B]; but under the pressure of the current litigation, this figure has been revised, firstly to 1 in 250,000, then 1 in 100,000, and lately to 1 in 34,000 [M1].

As customers who complain are still chased away by branch staff, and since a lot of people will just fail to notice one-off debits, our best guess is that the real figure is about 1 in 10,000. Thus, if an average customer uses an ATM once a week for 50 years, we would expect that about one in four customers will experience an ATM problem at some time in their lives.

Bankers are thus throwing away a lot of goodwill, and their failure to face up to the problem may undermine confidence in the payment system and contribute to unpopularity, public pressure and ultimately legislation. While they consider their response to this, they are not only under fire in the press and the courts, but are also saddled with systems which they built from components which were not understood, and whose administrative support requirements have almost never been adequately articulated. This is hardly the environment in which a clear headed and sensible strategy is likely to emerge.

The implications for equipment vendors

Equipment vendors will argue that real security expertise is only to be found in universities, government departments, one or two specialist consultancy firms, and in their design labs. Because of this skill shortage, only huge projects will have a capable security expert on hand during the whole of the development and implementation process. Some projects may get a short consultancy input, but the majority will have no specialised security effort at all. The only way in which the experts' knowhow can be brought to market is therefore in the form of products, such as hardware devices, software packages and training courses.

If this argument is accepted, then our research implies that vendors are currently selling the wrong products, and governments are encouraging this by certifying these products under schemes like ITSEC.

As we have seen, the suppliers' main failure is that they overestimate their customers' level of cryptologic and security design sophistication.

IBM's security products, such as the 3848 and the newer 4753, are a good case in point: they provide a fairly raw encryption capability, and leave the application designer to worry about protocols and to integrate the cryptographic facilities with application and system software.

This may enable IBM to claim that a 4753 will do any cryptographic function that is required, that it can handle both military and civilian security requirements and that it can support a wide range of security architectures [JDKLM]; but the hidden cost of this flexibility is that almost all their customers lack the skills to do a proper job, and end up with systems which have bugs.

A second problem is that those security functions which have to be implemented at the application level end up being neglected. For example, security modules provide a warning message if a decrypted key has the wrong parity, which would let the bank know that someone is experimenting with the system; but there is usually no mainframe software to relay this warning to anyone who can act on it.

The third reason why equipment designers should be on guard is that the threat environment is not constant, or even smoothly changing. In many countries, organised crime ignored ATMs for many years, and losses remained low; once they took an interest, the effect was dramatic [BAB]. In fact, we would not be too surprised if the Mafia were to build a keysearch machine to attack the zone keys used in ATM networks. This may well not happen, but banks and their suppliers should work out how to react if it does.

A fourth problem is that sloppy quality control can make the whole exercise pointless. A supplier of equipment whose purpose is essentially legal rather than military may at any time be the subject of an order for disclosure or discovery, and have his design notes, source code and test data seized for examination by hostile expert witnesses. If they find flaws, and the case is then lost, the supplier could face ruinous claims for damages from his client. This may be a more hostile threat environment than that faced by any military supplier, but the risk does not seem to be appreciated by the industry.

In any case, it appears that implementing secure computer systems using the available encryption products is beyond most organisations' capabilities, as indeed is maintaining and managing these systems once they have been installed. Tackling this problem will require:

- a system level approach to designing and evaluating security. This is the important question, which we will discuss in the next section
- a certification process which takes account of the human environment in which the system will operate. This is the urgent question.

The urgency comes from the fact that many companies and government departments will continue to buy whatever products have been recommended by the appropriate authority, and then, because they lack the skill to implement and manage the security features, they will use them to build systems with holes.

This outcome is a failure of the certification process. One would not think highly of an inspector who certified the Boeing 747 or the Sukhoi Su-26 for use as a basic trainer, as these aircraft take a fair amount of skill to fly. The aviation community understands this, and formalises it through a hierarchy of licences - from the private pilot's licence for beginners, through various commercial grades, to the airline licence which is a legal requirement for the captain of any scheduled passenger flight.

In the computer security community, however, this has not happened yet to any great extent. There are some qualifications (such as Certified Information Systems Auditor) which are starting to gain recognition, especially in the USA, but most computer security managers and staff cannot be assumed to have had any formal training in the subject.

There are basically three courses of action open to equipment vendors:

- to design products which can be integrated into systems, and thereafter maintained and managed, by computer staff with a realistic level of expertise
- to train and certify the client personnel who will implement the product into a system, and to provide enough continuing support to ensure that it gets maintained and managed adequately
- to supply their own trained and bonded personnel to implement, maintain and manage the system.

The ideal solution may be some combination of these. For example, a vendor might perform the implementation with its own staff; train the customer's staff to manage the system thereafter; and design the product so that the only maintenance possible is the replacement of complete units. However, vendors and their customers should be aware that both the second and third of the above options carry a significant risk that the security achieved will deteriorate over time under normal budgetary pressures.

Whatever the details, we would strongly urge that information security products should not be certified under schemes like ITSEC unless the manufacturer can show that both the system factors and the human factors have been properly considered. Certification must cover not just the hardware and software design, but also installation, training, maintenance, documentation and all the support that may be required by the applications and environment in which the product is licensed to be used.

The Wider Implications

As we have seen, security equipment designers and government evaluators have both concentrated on technical weaknesses, such as poor encryption algorithms and operating systems which could be vulnerable to trojan horse attacks. Banking systems do indeed have their share of such loopholes, but they do not seem to have contributed in any significant way to the crime figures.

The attacks which actually happened were made possible because the banks did not use the available products properly; due to lack of expertise, they made basic errors in system design, application programming and administration.

In short, the threat model was completely wrong. How could this have happened?

Why the threat model was wrong

During the 1980's, there was an industry wide consensus on the threat model, which was reinforced at conferences and in the literature. Designers concentrated on what could possibly happen rather than on what was likely to happen, and assumed that criminals would have the expertise, and use the techniques, of a government signals agency. More seriously, they assumed that implementers at customer sites would have either the expertise to design and build secure systems using the components they sold, or the common sense to call in competent consultants to help. This was just not the case.

So why were both the threat and the customers' abilities so badly misjudged?

The first error may be largely due to an uncritical acceptance of the conventional military wisdom of the 1970's. When ATMs were developed and a need for cryptographic expertise became apparent, companies imported this expertise from the government sector [C3]. The military model stressed secrecy, so secrecy of the PIN was made the cornerstone of the ATM system: technical efforts were directed towards ensuring it, and business and legal strategies were predicated on its being achieved. It may also be relevant that the early systems had only limited networking, and so the security design was established well before ATM networks acquired their present size and complexity.

Nowadays, however, it is clear that ATM security involves a number of goals, including controlling internal fraud, preventing external fraud, and arbitrating disputes fairly, even when the customer's home bank and the ATM raising the debit are in different countries. This was just not understood in the 1970's; and the need for fair arbitration in paticular seems to have been completely ignored.

The second error was probably due to fairly straightforward human factors. Many organisations have no computer security team at all, and those that do have a hard time finding it a home within the administrative structure. The internal audit department, for example, will resist being given any line management tasks, while the programming staff dislike anyone whose rôle seems to be making their job more difficult.

Security teams thus tend to be `reorganised' regularly, leading to a loss of continuity; a recent study shows, for example, that the average tenure of computer security managers at US government agencies is only seven months [H]. In the rare cases where a security department does manage to thrive, it usually has difficulties attracting and keeping good engineers, as they get bored once the initial development tasks have been completed.

These problems are not unknown to security equipment vendors, but they are more likely to flatter the customer and close the sale than to tell him that he needs help.

This leaves the company's managers as the only group with the motive to insist on good security. However, telling good security from bad is notoriously difficult, and many companies would admit that technical competence (of any kind) is hard to instil in managers, who fear that becoming specialised will sidetrack their careers.

Corporate politics can have an even worse effect, as we saw above: even where technical staff are aware of a security problem, they often keep quiet for fear of causing a powerful colleague to lose face.

Finally we come to the `consultants': most banks buy their consultancy services from a small number of well known firms, and value an `air of certainty and quality' over technical credentials. Many of these firms pretend to expertise which they do not possess, and cryptology is a field in which it is virtually impossible for an outsider to tell an expert from a charlatan. The author has seen a report on the security of a national ATM network switch, where the inspector (from an eminent firm of chartered accountants) completely failed to understand what encryption was, and under the heading of communications security remarked that the junction box was well enough locked up to keep vagrants out!

Confirmation of our analysis

It has recently become clear (despite the fog of official secrecy) that the military sector has suffered exactly the same kind of experiences that we described above. The most dramatic confirmation came at a workshop held in Cambridge in April 93 [M2], where a senior NSA scientist, having heard a talk by the author on some of these results, said that:

- the vast majority of security failures occur at the level of implementation detail
- the NSA is not cleverer than the civilian security community, just better informed of the threats. In particular, there are `platoons' of people whose career speciality is studying and assessing threats of the kind discussed here
- the threat profiles developed by the NSA for its own use are classified

This was encouraging, as it shows that our work is both accurate and important. However, with hindsight, it could have been predicted. Kahn, for example, attributes the Russian disasters of World War 1 to the fact that their soldiers found the more sophisticated army cipher systems too hard to use, and reverted to using simple systems which the Germans could solve without great difficulty [K1].

More recently, Price's survey of US Department of Defence organisations has found that poor implementation is the main security problem there [P]: although a number of systems use `trusted components', there are few, if any, operational systems which employ their features effectively. Indeed, it appears from his research that the availability of these components has had a negative effect, by fostering complacency: instead of working out a system's security requirements in a methodical way, designers just choose what they think is the appropriate security class of component and then regurgitate the description of this class as the security specification of the overall system.

The need for more emphasis on quality control is now gaining gradual acceptance in the military sector; the US Air Force, for example, is implementing the Japanese concept of `total quality management' in its information security systems [SSWDC]. However, there is still a huge vested interest in the old way of doing things; many millions have been invested in TCSEC and ITSEC compliant products, and this investment is continuing. A more pragmatic approach, based on realistic appraisal of threats and of organisational and other human factors, will take a long time to become approved policy and universal practice.

Nonetheless both our work, and its military confirmation, indicate that a change in how we do cryptology and computer security is needed, and there are a number of signs that this change is starting to get under way.

A New Security Paradigm?

As more people become aware of the shortcomings of traditional approaches to computer security, the need for new paradigms gets raised from time to time. In fact, there are now workshops on the topic [NSP], and an increasing number of journal papers make some kind of reference to it.

It is clear from our work that, to be effective, this change must bring about a change of focus. Instead of worrying about what might possibly go wrong, we need to make a systematic study of what is likely to; and it seems that the core security business will shift from building and selling `evaluated' products to an engineering discipline concerned with quality control processes within the client organisation.

When a paradigm shift occurs [K2], it is quite common for a research model to be imported from some other discipline in order to give structure to the newly emerging results. For example, Newton dressed up his dramatic results on mechanics in the clothing of Euclidean geometry, which gave them instant intellectual respectability; and although geometry was quickly superseded by calculus, it was a useful midwife at the birth of the new science. It also had a lasting influence in its emphasis on mathematical elegance and proof.

So one way for us to proceed would be to look around for alternative models which we might usefully import into the security domain. Here, it would seem that the relationship between secure systems and safety critical systems will be very important.

A new metaphor

Safety critical systems have been the subject of intensive study, and the field is in many ways more mature than computer security. There is also an interesting technical duality, in that while secure systems must do at most *X*, critical systems must do at least *X*; and while many secure systems must have the property that processes write up and read down, critical systems are the opposite in that they write down and read up. We might therefore expect that many of the concepts would go across, and again it is the US Air Force which has discovered this to be the case [JAJP]. The relationship between security and safety has also been investigated by other researchers [BMD].

There is no room here for a treatise on software engineering for safety critical systems, of which there are a number of introductory articles available [C4]. We will mention only four very basic points [M3]:

1. The specification should list all possible failure modes of the system. This should include every substantially new accident or incident which has ever been reported and which is relevant to the

equipment being specified.

2. The specification should make clear what strategy has been adopted to prevent each of these failure modes, or at least make them acceptably unlikely.

3. The specification should then explain in detail how each of these failure management strategies is implemented, including the consequences when each single component, subroutine or subassembly of the system itself fails. This explanation must be assessed by independent experts, and it must cover not just technical design factors, but training and operational issues too. If the procedure when an engine fails is to fly on with the other engine, then what skills does a pilot need to do this, and what are the procedures whereby these skills are acquired, kept current and tested?

4. The certification program must test whether the equipment can in fact be operated by people with the level of skill and experience assumed in the specification. It must also include a monitoring program whereby all incidents are reported to both the equipment manufacturer and the certification body.

These points tie in exactly with our findings (and with the NSA's stated experience). However, even a cursory comparison with the ITSEC programme shows that this has a long way to go. As we mentioned in the introduction, no-one seems so far to have attempted even the first stage of the safety engineering process for commercial cryptographic systems.

As for the other three stages, it is clear that ITSEC (and TCSEC) will have to change radically. Component-oriented security standards and architectures tend to ignore the two most important factors, which are the system aspect and the human element; in particular, they fail to ensure that the skills and performance required of various kinds of staff are included, together with the hardware and software, in the certification loop.

The competing philosophies

Within the field of critical systems, there are a number of competing approaches. The first is epitomised by railway signalling systems, and seeks either to provide multiple redundant interlocks or to base the safety features on the integrity of a kernel of hardware and software which can be subjected to formal verification [CW].

The second is the aviation paradigm which we introduced at the beginning of this article; here the quality engineering process is based on constant top level feedback and incremental improvement. This feedback also occurs at lower levels, with various distinct subsystems (pilot training, maintenance, airworthiness certification, traffic control, navigational aids, ...) interacting in fairly well understood ways with each other.

Of these two models, the first is more reductionist and the second more holist. They are not mutually exclusive (formal verification of avionics is not a bad thing, unless people then start to trust it too much); the main difference is one of system philosophy.

The most basic aspect of this is that in signalling systems, the system is in control; if the train driver falls asleep, or goes through a red light, the train will stop automatically. His task has been progressively deskilled until his main function is to see that the train stops precisely at the platform (and in some modern railways, even this task is performed automatically, with the result that driverless trains are beginning to enter service).

In civil aviation, on the other hand, the pilot remains firmly in command, and progress has made his job ever more complex and demanding. It was recently revealed, for example, that Boeing 747 autopilots have for 22 years been subject to erratic failures, which can result in the plane starting to roll.

Boeing's response was blunt: autopilots `are designed to assist and supplement the pilot's capabilities and not replace them', the company said [CR]. `This means our airplanes are designed so pilots are the final control authority and it means that a well trained crew is the first line of safety.'

The computer security implications

Both the railway and airline models find reflections in current security practice and research. The former model is dominant, due to the TCSEC/ITSEC emphasis on kernelisation and formal methods. In addition to the conventional multilevel secure evaluated products, kernelisation has been used at the application layer as well [A2] [C5].

Nonetheless, we must consider whether this is the right paradigm to adopt. Do we wish to make the computer security officer's job even more mechanical, and perhaps automate it entirely? This is the direction in which current trends seem to lead, and if our parallel with signalling systems is accurate, it is probably a blind alley; we should follow the aviation paradigm instead.

Another analogy is presented in [BGS], where it is argued that the traditional centralised model of security is like the old communist approach to economic management, and suffers from the same limitations. The authors there argue that to cope with a world of heterogeneous networks in which no single security policy is able to predominate, we need an infrastructure which enables information owners to control and trade their own property, rather than trusting everything to a centralised administrative structure.

This analogy from economics would, if developed, lead to somewhat similar conclusions to those which we draw from comparing railway signals with air traffic control systems. No doubt many other analogies will be explored over the next few years; the key point seems to be that, to be useful, a security metaphor should address not just the technical issues, but the organisational ones as well.

Conclusions

Designers of cryptographic systems have suffered from a lack of information about how their products fail in practice, as opposed to how they might fail in theory. This lack of feedback has led to a false threat model being accepted. Designers focussed on what could possibly go wrong, rather than on what was likely to; and many of their products are so complex and tricky to use that they are rarely used properly.

As a result, most security failures are due to implementation and management errors. One specific consequence has been a spate of ATM fraud, which has not just caused financial losses, but has also caused at least one miscarriage of justice and has eroded confidence in the UK banking system. There has also been a military cost; the details remain classified, but its existence has at last been admitted.

Our work also shows that component-level certification, as embodied in both the ITSEC and TCSEC programs, is unlikely to achieve its stated goals. This, too, has been admitted indirectly by the military (at least in the USA); and we would recommend that the next versions of these standards take much more account of the environments in which the components are to be used, and especially the system and human factors.

Most interesting of all, however, is the lesson that the bulk of computer security research and development activity is expended on activities which are of marginal relevance to real needs. A paradigm shift is underway, and a number of recent threads point towards a fusion of security with software engineering, or at the very least to an influx of software engineering ideas.

Our work also raises some very basic questions about goals, and about how the psychology of a design interacts with organisational structure. Should we aim to automate the security process, or enable it to be managed? Do we control or facilitate? Should we aim for monolithic systems, or devise strategies to cope with diversity? Either way, the tools and the concepts are becoming available. At least we should be aware that we have the choice.

Acknowledgement: I owe a significant debt to Karen Sparck Jones, who went through the manuscript of this paper and ruthlessly struck out all the jargon. Without her help, it would have been readable only by specialists.

Bibliography

A1

D Austin, ``Marking the Cards", in *Banking Technology*, Dec 91/Jan 92, pp 18 - 21

A2

RJ Anderson, ``UEPS - A Second Generation Electronic Wallet''. in *Computer Security - ESORICS 92*, Springer LNCS **648**, pp 411 - 418

B

M Buckler MP, letter to plaintiff's solicitor, 8 June 1992

BAB

``Card Fraud: Banking's Boom Sector'', in *Banking Automation Bulletin for Europe*, Mar 92, pp 1 - 5

BAN

M Burrows, M Abadi and RM Needham, *`A Logic of Authentication'*, DEC SRC Research Report **39**

BB

``Cash Dispenser Security'', *Barclays Briefing* (press release) 12/9/92

BGS

JA Bull, L Gong, K Sollins, ``Towards Security in an Open Systems Federation'', in *Proceedings of ESORICS 92*, Springer LNCS **648** pp 3 - 20

BMD

A Burns, JA McDermid, JE Dobson, *`On the meaning of safety and security'*, University of Newcastle upon Tyne Computer Laboratory **TR 382** (5/92)

C1

A Collins, ``Bank worker guilty of ATM fraud'', in *Sunday Times*, 22 Mar 1992

C2

A Collins, ``The Machines That Never Go Wrong'', in *Computer Weekly*, 27 June 1992, pp 24 - 25

C3

D Coppersmith, ``The Data Encryption Standard (DES) and its strength against attacks'', IBM Thomas J Watson Research Center technical report **RC 18613 (81421)**, 22 December 1992

C4

J Cullyer, ``Safety-critical systems'', in *Computing and Control Engineering Journal* **2** no 5 (Sep 91) pp 202 - 210

C5

B Christianson, ``Document Integrity in CSCW", in *Proc. Cambridge Workshop on Formal Methods* (1993, to appear)

CR

Boeing News Digest, quoted in usenet newsgroup `` `comp.risks' `` **14** no 5 (29 April 1993)

CW

J Cullyer, W Wong, ``Application of formal methods to railway signalling - a case study", in *Computing and Control Engineering Journal* **4** no 1 (Feb 93) pp 15 - 22

DP

DW Davies and WL Price, *`Security for Computer Networks'*, John Wiley and Sons 1984.

E

J Essinger, *`ATM Networks - Their Organisation, Security and Future'*, Elsevier 1987

GO

G Garon and R Outerbridge, ``DES Watch: An examination of the Sufficiency of the Data Encryption Standard for Financial Institution Information Security in the 1990's, in *Cryptologia*, **XV**, no. 3 (July 1991) pp 177 - 193

H

HJ Highland, ``Perspectives in Information Technology Security", in *Proceedings of the 1992 IFIP Congress, `Education and Society', IFIP A-13 vol II* (1992) pp 440 - 446

ITSEC

`Information Technology Security Evaluation Criteria', June 1991, EC document COM(90) 314

J1

RB Jack (chairman), *`Banking services: law and practice report by the Review Committee'*, HMSO, London, 1989

J2

K Johnson, ``One Less Thing to Believe In: Fraud at Fake Cash Machine", in *New York Times* 13 May 1993 p 1

JAJP

HL Johnson, C Arvin, E Jenkinson, R Pierce, ``Integrity and assurance of service protection in a large, multipurpose, critical system" in *proceedings of the 15th National Computer Security Conference*, NIST (1992) pp 252 - 261

JC

Dorothy Judd v Citibank, *435 NYS, 2d series, pp 210 - 212, 107 Misc.2d 526*

JDKLM

DB Johnson, GM Dolan, MJ Kelly, AV Le, SM Matyas, ``Common Cryptographic Architecture Application Programming Interface'', in *IBM Systems Journal* **30** no 2 (1991) pp 130 - 150

K1

D Kahn, *`The Codebreakers'*, Macmillan 1967

K2

TS Kuhn, *`The Structure of Scientific Revolutions'*, Chicago 1970

L1

B Lewis, ``How to rob a bank the cashcard way'', in *Sunday Telegraph* 25th April 1992 p 5

L2

D Lane, ``Where Cash is King'', in *Banking Technology*, October 1992, pp 38 - 41

M1

S McConnell, ``Barclays defends its cash machines'', in *The Times*, 7 November 1992

M2

R Morris, invited lecture given at Cambridge 1993 formal methods workshop (proceedings to appear)

M3

JA McDermid, ``Issues in the Development of Safety Critical Systems'', *public lecture, 3rd February 1993*

MB

McConville & others v Barclays Bank & others, High Court of Justice Queen's Bench Division 1992 ORB no.812

MBW

McConville & others v Barclays Bank & others *cit*, affidavit by D Whalley

MM

CH Meyer and SM Matyas, *`Cryptography: A New Dimension in Computer Data Security'*, John Wiley and Sons 1982.

N

I Newton, `Philosophiae Naturalis Principia Mathematica', University of California Press 1973

NSM

`Network security Module - Application Developer's Manual', Computer Security Associates, 1990

NSP

New Security Paradigms Workshop, 2-5 August 1993, proceedings to be published by the ACM.

P

WR Price, ``Issues to Consider When Using Evaluated Products to Implement Secure Mission Systems'', in *Proceedings of the 15th National Computer Security Conference*, National Institute of Standards and Technology (1992) pp 292 - 299

R

RA Rueppel, ``Criticism of ISO CD 11166 Banking: Key Management by Means of Asymmetric Algorithms'', in *Proceedings of 3rd Symposium of State and Progress of Research in Cryptography*, Fondazione Ugo Bordoni, Rome 1993

RM

R v Moon, *Hastings Crown Court*, Feb 92

RSH

R v Stone and Hider, *Winchester Crown Court* July 1991

S

A Stone, ``ATM cards & fraud'', *manuscript 1993*

SSWDC

L Sutterfield, T Schell, G White, K Doster and D Cuiskelly, ``A Model for the Measurement of Computer Security Posture'', in *Proceedings of the 15th National Computer Security Conference*, NIST (1992) pp 379 - 388

TCSEC

`Trusted Computer System Evaluation Criteria, US Department of Defense, 5200.28-STD, December 1985

VSM

`VISA Security Module Operations Manual', VISA, 1986

W1

G Welchman, *The Hut Six Story*, McGraw-Hill, 1982

W2

MA Wright, `Security Controls in ATM Systems', in *Computer Fraud and Security Bulletin*, November 1991, pp 11 - 14

W3

K Wong, `Data security - watch out for the new computer criminals", in *Computer Fraud and Security Bulletin*, April 1987, pp 7 - 13

About this document ...

Why Cryptosystems Fail

This paper was published by the ACM in 1st Conf.- Computer and Comm. Security '93 -- 11/93 -- VA, USA, ISBN 0-89791-629-8. Permission to copy all or part of this material is given provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission. 93/0011 ... \$1.50

This document was generated using the **LaTeX**2`HTML` translator Version 98.1p1 release (March 2nd, 1998)

The command line arguments were:
latex2html `-split 0 wcf.tex`.

The translation was initiated by Ross Anderson on 1999-06-08

Ross Anderson
1999-06-08

Why Cryptosystems Fail

Ross Anderson
University Computer Laboratory
Pembroke Street, Cambridge CB2 3QG
Email: rja14@cl.cam.ac.uk

Abstract

Designers of cryptographic systems are at a disadvantage to most other engineers, in that information on how their systems fail is hard to get: their major users have traditionally been government agencies, which are very secretive about their mistakes.

In this article, we present the results of a survey of the failure modes of retail banking systems, which constitute the next largest application of cryptology. It turns out that the threat model commonly used by cryptosystem designers was wrong: most frauds were not caused by cryptanalysis or other technical attacks, but by implementation errors and management failures. This suggests that a paradigm shift is overdue in computer security; we look at some of the alternatives, and see some signs that this shift may be getting under way.

1 Introduction

Cryptology, the science of code and cipher systems, is used by governments, banks and other organisations to keep information secure. It is a complex subject, and its national security overtones may invest it with a certain amount of glamour, but we should never forget that information security is at heart an engineering problem. The hardware and software products which are designed to solve it should in principle be judged in the same way as any other products: by their cost and effectiveness.

However, the practice of cryptology differs from, say, that of aeronautical engineering in a rather striking way: there is almost no public feedback about how cryptographic systems fail.

When an aircraft crashes, it is front page news. Teams of investigators rush to the scene, and the subsequent enquiries are conducted by experts from organisations with a wide range of interests - the carrier, the insurer, the manufacturer, the airline pilots' union, and the local aviation authority. Their findings are examined by journalists and politicians, discussed in pilots' messes, and passed on by flying instructors.

In short, the flying community has a strong and institutionalised learning mechanism. This is perhaps the main reason why, despite the inherent hazards of flying in large aircraft, which are maintained and piloted by fallible human beings, at hundreds of miles an hour through congested airspace, in bad weather and at night, the risk of being killed on an air journey is only about one in a million.

In the crypto community, on the other hand, there is no such learning mechanism. The history of the subject ([K1], [W1]) shows the same mistakes being made over and over again; in particular, poor management of codebooks and cipher machine procedures enabled many communication networks to be broken. Kahn relates, for example [K1, p 484], that Norway's rapid fall in the second world war was largely due to the fact that the British Royal Navy's codes had been solved by the German Beobachtungsdienst - using exactly the same techniques that the Royal Navy's own 'Room 40' had used against Germany in the previous war.

Since world war two, a curtain of silence has descended on government use of cryptography. This is not surprising, given not just the cold war, but also the reluctance of bureaucrats (in whatever organisation) to admit their failures. But it does put the cryptosystem designer at a severe disadvantage compared with engineers working in other disciplines; the post-war years are precisely the period in which modern cryptographic systems have been developed and brought into use. It is as if accident reports were only published for piston-engined aircraft, and the causes of all jet aircraft crashes were kept a state secret.

2 Automatic Teller Machines

To discover out how modern cryptosystems are vulnerable in practice, we have to study their use elsewhere. After government, the next biggest application is in banking, and evolved to protect automatic teller machines (ATMs) from fraud.

In some countries (including the USA), the banks have to carry the risks associated with new technology. Following a legal precedent, in which a bank customer's word that she had not made a withdrawal was found to outweigh the banks' experts' word that she must have done [JC], the US Federal Reserve passed regulations which require banks to refund all disputed transactions unless they can prove fraud by the customer [E]. This has led to some minor abuse - misrepresentations by customers are estimated to cost the average US bank about $15,000 a year [W2] - but it has helped promote the development of security technologies such as cryptology and video.

In Britain, the regulators and courts have not yet been so demanding, and despite a parliamentary commission of enquiry which found that the PIN system was insecure [J1], bankers simply deny that their systems are ever at fault. Customers who complain about debits on their accounts for which they were not responsible - so-called 'phantom withdrawals' - are told that they are lying, or mistaken, or that they must have been defrauded by their friends or relatives.

The most visible result in the UK has been a string of court cases, both civil and criminal. The pattern which emerges leads us to suspect that there may have been a number of miscarriages of justice over the years.

- A teenage girl in Ashton under Lyme was convicted in 1985 of stealing £40 from her father. She pleaded guilty on the advice of her lawyers that she had no defence, and then disappeared; it later turned out that there had been never been a theft, but merely a clerical error by the bank [MBW]
- A Sheffield police sergeant was charged with theft in November 1988 and suspended for almost a year after a phantom withdrawal took place on a card he had confiscated from a suspect. He was lucky in that his colleagues tracked down the lady who had made the transaction after the disputed one; her eyewitness testimony cleared him
- Charges of theft against an elderly lady in Plymouth were dropped after our enquiries showed that the bank's computer security systems were a shambles
- In East Anglia alone, we are currently advising lawyers in two cases where people are awaiting trial for alleged thefts, and where the circumstances give reason to believe that 'phantom withdrawals' were actually to blame.

Finally, in 1992, a large class action got underway in the High Court in London [MB], in which hundreds of plaintiffs seek to recover damages from various banks and building societies. We were retained by the plaintiffs to provide expert advice, and accordingly conducted some research during 1992 into the actual and possible failure modes of automatic teller machine systems. This involved interviewing former bank employees and criminals, analysing statements from plaintiffs and other victims of ATM fraud, and searching the literature. We were also able to draw on experience gained during the mid-80's on designing cryptographic equipment for the financial sector, and advising clients overseas on its use.

We shall now examine some of the ways in which ATM systems have actually been defrauded. We will then compare them with how the designers thought their products might in theory be vulnerable, and see what lessons can be drawn. Some material has had to be held back for legal reasons, and in particular we do not identify all the banks whose mistakes we discuss. This information should be provided by witnesses at trial, and its absence here should have no effect on the points we wish to make.

3 How ATM Fraud Takes Place

We will start with some simple examples which indicate the variety of frauds that can be carried out without any great technical sophistication, and the bank operating procedures which let them happen. For the time being, we may consider that the magnetic strip on the customer's card contains only his account number, and that his personal identification number (PIN) is derived by encrypting this account number and taking four digits from the result. Thus the ATM must be able to perform this encryption operation, or to check the PIN in some other way (such as by an online enquiry).

3.1 Some simple examples

1. Many frauds are carried out with some inside knowledge or access, and ATM fraud turns out to be no exception. Banks in the English speaking world dismiss about one percent of their staff every year for disciplinary reasons, and many of these sackings are for petty thefts in which ATMs can easily be involved. A bank with 50,000 staff, which issued cards and PINs through the branches rather than by post, might expect about two incidents per business day of staff stealing cards and PINs.

    - In a recent case, a housewife from Hastings, England, had money stolen from her account by a bank clerk who issued an extra card for it. The bank's systems not only failed to prevent this, but also had the feature that whenever a cardholder got a statement from an ATM, the items on it would not subsequently appear on the full statements sent to the account address. This enabled the clerk to see to it that she did not get any statement showing the thefts he had made from her account.

      This was one of the reasons he managed to make 43 withdrawals of £200 each; the other was that when she did at last complain, she was not believed. In fact she was subjected to harrassment by the bank, and the thief was only discovered because he suffered an attack of conscience and owned up [RM].
    - Technical staff also steal clients' money, knowing that complaints will probably be ignored. At one bank in Scotland, a maintenance engineer fitted an ATM with a handheld computer, which recorded customers' PINs and account numbers. He then made up counterfeit cards and looted their accounts [C1] [C2]. Again, customers who complained were stonewalled; and the bank was

publicly criticised for this by one of Scotland's top law officers.

- One bank issues tellers with cards with which they can withdraw money from branch ATMs and debit any customer account. This may be convenient when the teller station cash runs out, but could lead the staff into temptation.
- One bank had a well managed system, in which the information systems, electronic banking and internal audit departments cooperated to enforce tight dual control over unissued cards and PINs in the branches. This kept annual theft losses down, until one day a protegé of the deputy managing director sent a circular to all branches announcing that to cut costs, a number of dual control procedures were being abolished, including that on cards and PINs. This was done without consultation, and without taking any steps to actually save money by reducing staff. Losses increased tenfold; but managers in the affected departments were unwilling to risk their careers by making a fuss. This seems to be a typical example of how computer security breaks down in real organisations.

Most thefts by staff show up as phantom withdrawals at ATMs in the victim's neighbourhood. English banks maintain that a computer security problem would result in a random distribution of transactions round the country, and as most disputed withdrawals happen near the customer's home or place of work, these must be due to cardholder negligence [BB]. Thus the pattern of complaints which arises from thefts by their own staff only tends to reinforce the banks' complacency about their systems.

2. Outsiders have also enjoyed some success at attacking ATM systems.

    - In a recent case at Winchester Crown Court in England [RSH], two men were convicted of a simple but effective scam. They would stand in ATM queues, observe customers' PINs, pick up the discarded ATM tickets, copy the account numbers from the tickets to blank cards, and use these to loot the customers' accounts.

      This trick had been used (and reported) several years previously at a bank in New York. There the culprit was an ATM technician, who had been fired, and who managed to steal over $80,000 before the bank saturated the area with security men and caught him in the act.

      These attacks worked because the banks printed the full account number on the ATM ticket, and because there was no cryptographic redundancy on the magnetic strip. One might have thought that the New York lesson would have been learned, but no: in England, the bank which had been the main victim in the Winchester case only stopped printing the full account number in mid 1992, after the author replicated the fraud on television to warn the public of the risk. Another bank continued printing it into 1993, and was pilloried by journalists who managed to forge a card and use it [L1].
    - Another technical attack relies on the fact that most ATM networks do not encrypt or authenticate the authorisation response to the ATM. This means that an attacker can record a 'pay' response from the bank to the machine, and then keep on replaying it until the machine is empty. This technique, known as 'jackpotting', is not limited to outsiders - it appears to have been used in 1987 by a bank's operations staff, who used network control devices to jackpot ATMs where accomplices were waiting.
    - Another bank's systems had the feature that when a telephone card was entered at an ATM, it believed that the previous card had been inserted again. Crooks stood in line, observed customers' PINs, and helped themselves. This shows how even the most obscure programming error can lead to serious problems.
    - Postal interception is reckoned to account for 30% of all UK payment card losses [A1], but most banks' postal control procedures are dismal. For example, in February 1992 the author asked for an increased card limit: the bank sent not one, but two, cards and PINs through the post. These cards arrived only a few days after intruders had got hold of our apartment block's mail and torn it up looking for valuables.

      It turned out that this bank did not have the systems to deliver a card by registered post, or to send it to a branch for collection. Surely they should have noticed that many of their Cambridge customers live in colleges, student residences and apartment buildings which have no secure postal deliveries; and that most of the new students open bank accounts at the start of the academic year in October, when large numbers of cards and PIN mailers are left lying around on staircases and in pigeonholes.
    - Test transactions have been another source of trouble. There was a feature on one make of ATM which would output ten banknotes when a 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. These went on until all the banks using the machine put in a software patch to disable the transaction.
    - The fastest growing modus operandi is to use false terminals to collect customer card and PIN data. Attacks of this kind were first reported from the USA in 1988; there, crooks built a vending machine which would accept any card and PIN, and dispense a packet of cigarettes. They put their invention in a shopping mall, and harvested PINs and magnetic strip data by modem. A more recent instance of this in Connecticut got substantial press publicity [J2], and the trick has spread to other countries too: in 1992, criminals set up a market stall in High Wycombe, England, and customers who wished to pay for goods by credit card were asked to swipe the card and enter the PIN at a terminal which was in fact hooked up to a PC. At the time of writing, British banks had still not warned their customers of this threat.

3. The point of using a four-digit PIN is that someone who finds or steals another person's ATM card has a chance of only one in ten thousand of guessing the PIN, and if only three attempts are allowed, then the likelihood of a stolen card being misused should be less than one in 3,000. However, some banks have managed to reduce the diversity of a four-digit PIN to much less than 10,000. For example:

   - They may have a scheme which enables PINs to be checked by offline ATMs and point-of-sale devices without these devices having a full encryption capability. For example, customers of one bank get a credit card PIN with digit one plus digit four equal to digit two plus digit three, and a debit card PIN with one plus three equals two plus four. This means that crooks could use stolen cards in offline devices by entering a PIN such as 4455.
   - In early 1992, another bank sent its cardholders a letter warning them of the dangers of writing their PIN on their card, and suggested instead that they conceal the PIN in the following way and write it down on a distinctive piece of squared cardboard, which was designed to be kept alongside the ATM card in a wallet or purse.

     Suppose your PIN is 2256. Choose a four-letter word, say '`blue`'. Write these four letters down in the second, second, fifth and sixth columns of the card respectively:

1	2	3	4	5	6	7	8	9	0
	b								
	l								
				u					
					e				

     Now fill up the empty boxes with random letters. Easy, isn't it? Of course, there may be only about two dozen four-letter words which can be made up using a given grid of random letters, so a thief's chance of being able to use a stolen card has just increased from 1 in 3,333 to 1 in 8.
   - One small institution issued the same PIN to all its customers, as a result of a simple programming error. In yet another, a programmer arranged things so that only three different PINs were issued, with a view to forging cards by the thousand. In neither case was the problem detected until some considerable time had passed: as the live PIN mailers were subjected to strict handling precautions, no member of staff ever got hold of more than his own personal account mailer.

4. Some banks do not derive the PIN from the account number by encryption, but rather chose random PINs (or let the customers choose them) and then encrypt them for storage. Quite apart from the risk that customers may choose PINs which are easy to guess, this has a number of technical pitfalls.

   - Some banks hold the encrypted PINs on a file. This means that a programmer might observe that the encrypted version of his own PIN is (say) `132AD6409BCA4331`, and search the database for all other accounts with the same PIN.
   - One large UK bank even wrote the encrypted PIN to the card strip. It took the criminal fraternity fifteen years to figure out that you could change the account number on your own card's magnetic strip to that of your target, and then use it with your own PIN to loot his account.

     In fact, the Winchester pair used this technique as well, and one of them wrote a document about it which appears to have circulated in the UK prison system [S]; and there are currently two other men awaiting trial for conspiring to defraud this bank by forging cards.

   For this reason, VISA recommends that banks should combine the customer's account number with the PIN before encryption [VSM]. Not all of them do.

5. Despite all these horrors, Britain is by no means the country worst affected by card forgery. That dubious honour goes to Italy [L2], where losses amount to almost 0.5% of ATM turnover. Banks there are basically suffering from two problems.

   - The first is a plague of bogus ATMs - devices which look like real ATMs, and may even be real ATMs, but which are programmed to capture customers' card and PIN data. As we saw above, this is nothing new and should have been expected.
   - The second is that Italy's ATMs are generally offline. This means that anyone can open an account, get a card and PIN, make several dozen copies of the card, and get accomplices to draw cash from a number of different ATMs at the same time. This is also nothing new; it was a favourite modus operandi in Britain in the early 1980's [W3].

3.2 More complex attacks

The frauds which we have described so far have all been due to fairly simple errors of implementation and operation. Security researchers have tended to consider such blunders uninteresting, and have therefore concentrated on attacks which exploit more subtle technical weaknesses. Banking systems have a number of these weaknesses too.

Although high-tech attacks on banking systems are rare, they are of interest from the public policy point of view, as government initiatives such as the EC's Information Technology Security Evaluation Criteria [ITSEC] aim to develop a pool of evaluated products which have been certified free of known technical loopholes.

The basic assumptions behind this program are that implementation and operation will be essentially error-free, and that attackers will possess the technical skills which are available in a government signals security agency. It would therefore seem to be more relevant to military than civilian systems, although we will have more to say on this later.

In order to understand how these sophisticated attacks might work, we must look at banking security systems in a little more detail.

3.2.1 How ATM encryption works

Most ATMs operate using some variant of a system developed by IBM, which is documented in [MM]. This uses a secret key, called the 'PIN key', to derive the PIN from the account number, by means of a published algorithm known as the Data Encryption Standard, or DES. The result of this operation is called the 'natural PIN'; an offset can be added to it in order to give the PIN which the customer must enter. The offset has no real cryptographic function; it just enables customers to choose their own PIN. Here is an example of the process:

```
Account number:          8807012345691715
PIN key:                 FEFEFEFEFEFEFEFE
Result of DES:           A2CE126C69AEC82D
Result decimalised:      0224126269042823
Natural PIN:             0224
Offset:                  6565
Customer PIN:            6789
```

It is clear that the security of the system depends on keeping the PIN key absolutely secret. The usual strategy is to supply a 'terminal key' to each ATM in the form of two printed components, which are carried to the branch by two separate officials, input at the ATM keyboard, and combined to form the key. The PIN key, encrypted under this terminal key, is then sent to the ATM by the bank's central computer.

If the bank joins a network, so that customers of other banks can use its ATMs, then the picture becomes more complex still. 'Foreign' PINs must be encrypted at the ATM using a 'working' key it shares with its own bank, where they are decrypted and immediately re-encrypted using another working key shared with the card issuing bank.

These working keys in turn have to be protected, and the usual arrangement is that a bank will share a 'zone key' with other banks or with a network switch, and use this to encrypt fresh working keys which are set up each morning. It may also send a fresh working key every day to each of its ATMs, by encrypting it under the ATM's terminal key.

A much fuller description of banking security systems can be found in books such as [DP] and [MM], and in equipment manuals such as [VSM] and [NSM]. All we really need to know is that a bank has a number of keys which it must keep secret. The most important of these is of course the PIN key, as anyone who gets hold of this can forge a card for any customer's account; but other keys (such as terminal keys, zone keys and working keys) could also be used, together with a wiretap, to find out customer PINs in large numbers.

Keeping keys secret is only part of the problem. They must also be available for use at all times by authorised processes. The PIN key is needed all the time to verify transactions, as are the current working keys; the terminal keys and zone keys are less critical, but are still used once a day to set up new working keys.

The original IBM encryption products, such as PCF and the 3848, did not solve the problem: they only did the encryption step, and left the other manipulations to a mainframe computer program, which each bank had to write anew for itself. Thus the security depended on the skill and integrity of each bank's system development and maintenance staff.

The standard approach nowadays is to use a device called a security module. This is basically a PC in a safe, and it is programmed to manage all the bank's keys and PINs in such a way that the mainframe programmers only ever see a key or PIN in encrypted form. Banks which belong to the VISA and Mastercard ATM networks are supposed to use security modules, in order to prevent any bank customer's PIN becoming known to a programmer working for another bank (the Mastercard security requirements are quoted in [MM]; for VISA see [VSM]).

3.2.2 Problems with encryption products

In practice, there are a number of problems with encryption products, whether the old 3848s or the security modules now recommended by banking organisations. No full list of these problems, whether actual or potential, appears to have been published anywhere, but they include at least the following which have come to our notice:

1. Although VISA and Mastercard have about 10,000 member banks in the USA and at least 1,000 of these do their own processing, enquiries to security module salesmen reveal that only 300 of these processing centres had actually bought and installed these devices by late 1990. The first problem is thus that the hardware version of the product does not get bought at all, either because it is felt to be too expensive, or because it seems to be too difficult and time-consuming to install, or because it was not supplied by IBM (whose own security module product, the 4753, only became available in 1990). Where a bank has no security modules, the PIN encryption functions will typically be performed in software, with a number of undesirable consequences.

   - The first, and obvious, problem with software PIN encryption is that the PIN key can be found without too much effort by system programmers. In IBM's product, PCF, the manual even tells how to do this. Once armed with the PIN key, programmers can easily forge cards; and even if the bank installs security modules later, the PIN key is so useful for debugging the systems which support ATM networking that knowledge of it is likely to persist among the programming staff for years afterward.
   - Programmers at one bank did not even go to the trouble of setting up master keys for its encryption software. They just directed the key pointers to an area of low memory which is always zero at system startup. The effect of this was that the live and test systems could use the same cryptographic key dataset, and the bank's technicians found that they could work out customer PINs on their test equipment. Some of them used to charge the local underworld to calculate PINs on

stolen cards; when the bank's security manager found that this was going on, he was killed in a road accident (of which the local police conveniently lost the records). The bank has not bothered to send out new cards to its customers.

2. The 'buy-IBM-or-else' policy of many banks has backfired in more subtle ways. One bank had a policy that only IBM 3178 terminals could be purchased, but the VISA security modules they used could not talk to these devices (they needed DEC VT 100s instead). When the bank wished to establish a zone key with VISA using their security module, they found they had no terminal which would drive it. A contractor obligingly lent them a laptop PC, together with software which emulated a VT100. With this the various internal auditors, senior managers and other bank dignitaries duly created the required zone keys and posted them off to VISA.

   However, none of them realised that most PC terminal emulation software packages can be set to log all the transactions passing through, and this is precisely what the contractor did. He captured the clear zone key as it was created, and later used it to decrypt the bank's PIN key. Fortunately for them (and VISA), he did this only for fun and did not plunder their network (or so he claims).

3. Not all security products are equally good, and very few banks have the expertise to tell the good ones from the mediocre.

   - The security module's software may have trapdoors left for the convenience of the vendor's engineers. We only found this out because one bank had no proper ATM test environment; when it decided to join a network, the vendor's systems engineer could not get the gateway working, and, out of frustration, he used one of these tricks to extract the PIN key from the system, in the hope that this would help him find the problem. The existence of such trapdoors makes it impossible to devise effective control procedures over security modules, and we have so far been lucky that none of these engineers have tried to get into the card forgery business (or been forced to cooperate with organised crime).
   - Some brands of security module make particular attacks easier. Working keys may, for example, be generated by encrypting a time-of-day clock and thus have only 20 bits of diversity rather than the expected 56. Thus, according to probability theory, it is likely that once about 1,000 keys have been generated, there will be two of them which are the same. This makes possible a number of subtle attacks in which the enemy manipulates the bank's data communications so that transactions generated by one terminal seem to be coming from another.
   - A security module's basic purpose is to prevent programmers, and staff with access to the computer room, from getting hold of the bank's cryptographic keys. However, the 'secure' enclosure in which the module's electronics is packaged can often be penetrated by cutting or drilling. The author has even helped a bank to do this, when it lost the physical key for its security modules.
   - A common make of security module implements the tamper-protection by means of wires which lead to the switches. It would be trivial for a maintenance engineer to cut these, and then next time he visited that bank he would be able to extract clear keys.
   - Security modules have their own master keys for internal use, and these keys have to backed up somewhere. The backup is often in an easily readable form, such as PROM chips, and these may need to be read from time to time, such as when transferring control over a set of zone and terminal keys from one make of security module to another. In such cases, the bank is competely at the mercy of the experts carrying out the operation.
   - ATM design is also at issue here. Some older makes put the encryption in the wrong place - in the controller rather than in the dispenser itself. The controller was intended to sit next to the dispenser inside a branch, but many ATMs are no longer anywhere near a bank building. One UK university had a machine on campus which sent clear PINs and account data down a phone line to a controller in its mother branch, which is several miles away in town. Anyone who borrowed a datascope and used it on this line could have forged cards by the thousand.

4. Even where one of the better products is purchased, there are many ways in which a poor implementation or sloppy operating procedures can leave the bank exposed.

   - Most security modules return a whole range of response codes to incoming transactions. A number of these, such as 'key parity error' [VSM] give advance warning that a programmer is experimenting with a live module. However, few banks bother to write the device driver software needed to intercept and act on these warnings.
   - We know of cases where a bank subcontracted all or part of its ATM system to a 'facilities management' firm, and gave this firm its PIN key. There have also been cases where PIN keys have been shared between two or more banks. Even if all bank staff could be trusted, outside firms may not share the banks' security culture: their staff are not always vetted, are not tied down for life with cheap mortgages, and are more likely to have the combination of youth, low pay, curiosity and recklessness which can lead to a novel fraud being conceived and carried out.
   - Key management is usually poor. We have experience of a maintenance engineer being given both of the PROMs in which the security module master keys are stored. Although dual control procedures existed in theory, the staff had turned over since the PROMs were last used, and so no-one had any idea what to do. The engineer could not only have forged cards; he could have walked off with the PROMs and shut down all the bank's ATM operations.
   - At branch level, too, key management is a problem. As we have seen, the theory is that two

bankers type in one key component each, and these are combined to give a terminal master key; the PIN key, encrypted under this terminal master key, is then sent to the ATM during the first service transaction after maintenance.

  If the maintenance engineer can get hold of both the key components, he can decrypt the PIN key and forge cards. In practice, the branch managers who have custody of the keys are quite happy to give them to him, as they don't like standing around while the machine is serviced. Furthermore, entering a terminal key component means using a keyboard, which many older managers consider to be beneath their dignity.

  - We have accounts of keys being kept in open correspondence files, rather than being locked up. This applies not just to ATM keys, but also to keys for interbank systems such as SWIFT, which handles transactions worth billions. It might be sensible to use initialisation keys, such as terminal keys and zone keys, once only and then destroy them.
  - Underlying many of these control failures is poor design psychology. Bank branches (and computer centres) have to cut corners to get the day's work done, and only those control procedures whose purpose is evident are likely to be strictly observed. For example, sharing the branch safe keys between the manager and the accountant is well understood: it protects both of them from having their families taken hostage. Cryptographic keys are often not packaged in as user-friendly a way, and are thus not likely to be managed as well. Devices which actually look like keys (along the lines of military crypto ignition keys) may be part of the answer here.
  - We could write at great length about improving operational procedures (this is not a threat!), but if the object of the exercise is to prevent any cryptographic key from falling into the hands of someone who is technically able to abuse it, then this should be stated as an explicit objective in the manuals and training courses. 'Security by obscurity' often does more harm than good.

5. Cryptanalysis may be one of the less likely threats to banking systems, but it cannot be completely ruled out.

  - Some banks (including large and famous ones) are still using home-grown encryption algorithms of a pre-DES vintage. One switching network merely 'scrambled' data blocks by adding a constant to them; this went unprotested for five years, despite the network having over forty member banks - all of whose insurance assessors, auditors and security consultants presumably read through the system specification.
  - In one case, the two defendants tried to entice a university student into helping them break a bank's proprietary algorithm. This student was studying at a maths department where teaching and research in cryptology takes place, so the skills and the reference books were indeed available. Fortunately for the bank, the student went to the police and turned them in.
  - Even where a 'respectable' algorithm is used, it may be implemented with weak parameters. For example, banks have implemented RSA with key sizes between 100 and 400 bits, despite the fact that they key needs to be at least 500 bits to give any real margin of security.
  - Even with the right parameters, an algorithm can easily be implemented the wrong way. We saw above how writing the PIN to the card track is useless, unless the encryption is salted with the account number or otherwise tied to the individual card; there are many other subtle errors which can be made in designing cryptographic protocols, and the study of them is a whole discipline of itself [BAN]. In fact, there is open controversy about the design of a new banking encryption standard, ISO 11166, which is already in use by some 2,000 banks worldwide [R].
  - It is also possible to find a DES key by brute force, by trying all the possible encryption keys until you find the one which the target bank uses. The protocols used in international networks to encrypt working keys under zone keys make it easy to attack a zone key in this way: and once this has been solved, all the PINs sent or received by that bank on the network can be decrypted.

    A recent study by researchers at a Canadian bank [GO] concluded that this kind of attack would now cost about £30,000 worth of specialist computer time per zone key. It follows that it is well within the resources of organised crime, and could even be carried out by a reasonably well heeled individual.

    If, as seems likely, the necessary specialist computers have been built by the intelligence agencies of a number of countries, including countries which are now in a state of chaos, then there is also the risk that the custodians of this hardware could misuse it for private gain.

3.2.3 The consequences for bankers

The original goal of ATM crypto security was that no systematic fraud should be possible without the collusion of at least two bank staff [NSM]. Most banks do not seem to have achieved this goal, and the reasons have usually been implementation blunders, ramshackle administration, or both.

The technical threats described in section 3.2.2 above are the ones which most exercised the cryptographic equipment industry, and which their products were designed to prevent. However, only two of the cases in that section actually resulted in losses, and both of those can just as easily be classed as implementation failures.

The main technical lessons for bankers are that competent consultants should have been hired, and much greater emphasis should have been placed on quality control. This is urgent for its own sake: for in addition to fraud, errors also cause a significant number of disputed ATM transactions.

All systems of any size suffer from program bugs and operational blunders: banking systems are certainly no exception, as anyone who has worked in the industry will be aware.

Branch accounting systems tend to be very large and complex, with many interlocking modules which have evolved over decades. Inevitably, some transactions go astray: debits may get duplicated or posted to the wrong account.

This will not be news to financial controllers of large companies, who employ staff to reconcile their bank accounts. When a stray debit appears, they demand to see a voucher for it, and get a refund from the bank when this cannot be produced. However, the ATM customer with a complaint has no such recourse; most bankers outside the USA just say that their systems are infallible.

This policy carries with it a number of legal and administrative risks. Firstly, there is the possibility that it might amount to an offence, such as conspiracy to defraud; secondly, it places an unmeetable burden of proof on the customer, which is why the US courts struck it down [JC], and courts elsewhere may follow their lead; thirdly, there is a moral hazard, in that staff are encouraged to steal by the knowledge that they are unlikely to be caught; and fourthly, there is an intelligence failure, as with no central records of customer complaints it is not possible to monitor fraud patterns properly.

The business impact of ATM losses is therefore rather hard to quantify. In the UK, the Economic Secretary to the Treasury (the minister responsible for bank regulation) claimed in June 1992 that errors affected at most two ATM transactions out of the three million which take place every day [B]; but under the pressure of the current litigation, this figure has been revised, firstly to 1 in 250,000, then 1 in 100,000, and lately to 1 in 34,000 [M1].

As customers who complain are still chased away by branch staff, and since a lot of people will just fail to notice one-off debits, our best guess is that the real figure is about 1 in 10,000. Thus, if an average customer uses an ATM once a week for 50 years, we would expect that about one in four customers will experience an ATM problem at some time in their lives.

Bankers are thus throwing away a lot of goodwill, and their failure to face up to the problem may undermine confidence in the payment system and contribute to unpopularity, public pressure and ultimately legislation. While they consider their response to this, they are not only under fire in the press and the courts, but are also saddled with systems which they built from components which were not understood, and whose administrative support requirements have almost never been adequately articulated. This is hardly the environment in which a clear headed and sensible strategy is likely to emerge.

3.3 The implications for equipment vendors

Equipment vendors will argue that real security expertise is only to be found in universities, government departments, one or two specialist consultancy firms, and in their design labs. Because of this skill shortage, only huge projects will have a capable security expert on hand during the whole of the development and implementation process. Some projects may get a short consultancy input, but the majority will have no specialised security effort at all. The only way in which the experts' knowhow can be brought to market is therefore in the form of products, such as hardware devices, software packages and training courses.

If this argument is accepted, then our research implies that vendors are currently selling the wrong products, and governments are encouraging this by certifying these products under schemes like ITSEC.

As we have seen, the suppliers' main failure is that they overestimate their customers' level of cryptologic and security design sophistication.

IBM's security products, such as the 3848 and the newer 4753, are a good case in point: they provide a fairly raw encryption capability, and leave the application designer to worry about protocols and to integrate the cryptographic facilities with application and system software.

This may enable IBM to claim that a 4753 will do any cryptographic function that is required, that it can handle both military and civilian security requirements and that it can support a wide range of security architectures [JDKLM]; but the hidden cost of this flexibility is that almost all their customers lack the skills to do a proper job, and end up with systems which have bugs.

A second problem is that those security functions which have to be implemented at the application level end up being neglected. For example, security modules provide a warning message if a decrypted key has the wrong parity, which would let the bank know that someone is experimenting with the system; but there is usually no mainframe software to relay this warning to anyone who can act on it.

The third reason why equipment designers should be on guard is that the threat environment is not constant, or even smoothly changing. In many countries, organised crime ignored ATMs for many years, and losses remained low; once they took an interest, the effect was dramatic [BAB]. In fact, we would not be too surprised if the Mafia were to build a keysearch machine to attack the zone keys used in ATM networks. This may well not happen, but banks and their suppliers should work out how to react if it does.

A fourth problem is that sloppy quality control can make the whole exercise pointless. A supplier of equipment whose purpose is essentially legal rather than military may at any time be the subject of an order for disclosure or discovery, and have his design notes, source code and test data seized for examination by hostile expert witnesses. If they find flaws, and the case is then lost, the supplier could face ruinous claims for damages from his client. This may be a more hostile threat environment than that faced by any military supplier, but the risk does not seem to be appreciated by the industry.

In any case, it appears that implementing secure computer systems using the available encryption products is beyond most organisations' capabilities, as indeed is maintaining and managing these systems once they have been installed. Tackling this problem will require:

- a system level approach to designing and evaluating security. This is the important question, which we will discuss in the next section
- a certification process which takes account of the human environment in which the system will operate. This is the urgent question.

The urgency comes from the fact that many companies and government departments will continue to buy whatever products have been recommended by the appropriate authority, and then, because they lack the skill to implement and manage the security features, they will use them to build systems with holes.

This outcome is a failure of the certification process. One would not think highly of an inspector who certified the Boeing 747 or the Sukhoi Su-26 for use as a basic trainer, as these aircraft take a fair amount of skill to fly. The aviation community understands this, and formalises it through a hierarchy of licences - from the private pilot's licence for beginners, through various commercial grades, to the airline licence which is a legal requirement for the captain of any scheduled passenger flight.

In the computer security community, however, this has not happened yet to any great extent. There are some qualifications (such as Certified Information Systems Auditor) which are starting to gain recognition, especially in the USA, but most computer security managers and staff cannot be assumed to have had any formal training in the subject.

There are basically three courses of action open to equipment vendors:

- to design products which can be integrated into systems, and thereafter maintained and managed, by computer staff with a realistic level of expertise
- to train and certify the client personnel who will implement the product into a system, and to provide enough continuing support to ensure that it gets maintained and managed adequately
- to supply their own trained and bonded personnel to implement, maintain and manage the system.

The ideal solution may be some combination of these. For example, a vendor might perform the implementation with its own staff; train the customer's staff to manage the system thereafter; and design the product so that the only maintenance possible is the replacement of complete units. However, vendors and their customers should be aware that both the second and third of the above options carry a significant risk that the security achieved will deteriorate over time under normal budgetary pressures.

Whatever the details, we would strongly urge that information security products should not be certified under schemes like ITSEC unless the manufacturer can show that both the system factors and the human factors have been properly considered. Certification must cover not just the hardware and software design, but also installation, training, maintenance, documentation and all the support that may be required by the applications and environment in which the product is licensed to be used.

4 The Wider Implications

As we have seen, security equipment designers and government evaluators have both concentrated on technical weaknesses, such as poor encryption algorithms and operating systems which could be vulnerable to trojan horse attacks. Banking systems do indeed have their share of such loopholes, but they do not seem to have contributed in any significant way to the crime figures.

The attacks which actually happened were made possible because the banks did not use the available products properly; due to lack of expertise, they made basic errors in system design, application programming and administration.

In short, the threat model was completely wrong. How could this have happened?

4.1 Why the threat model was wrong

During the 1980's, there was an industry wide consensus on the threat model, which was reinforced at conferences and in the literature. Designers concentrated on what could possibly happen rather than on what was likely to happen, and assumed that criminals would have the expertise, and use the techniques, of a government signals agency. More seriously, they assumed that implementers at customer sites would have either the expertise to design and build secure systems using the components they sold, or the common sense to call in competent consultants to help. This was just not the case.

So why were both the threat and the customers' abilities so badly misjudged?

The first error may be largely due to an uncritical acceptance of the conventional military wisdom of the 1970's. When ATMs were developed and a need for cryptographic expertise became apparent, companies imported this expertise from the government sector [C3]. The military model stressed secrecy, so secrecy of the PIN was made the cornerstone of the ATM system: technical efforts were directed towards ensuring it, and business and legal strategies were predicated on its being achieved. It may also be relevant that the early systems had only limited networking, and so the security design was established well before ATM networks acquired their present size and complexity.

Nowadays, however, it is clear that ATM security involves a number of goals, including controlling internal fraud, preventing external fraud, and arbitrating disputes fairly, even when the customer's home bank and the ATM raising the debit are in different countries. This was just not understood in the 1970's; and the need for fair arbitration in paticular seems to have been completely ignored.

The second error was probably due to fairly straightforward human factors. Many organisations have no computer security team at all, and those that do have a hard time finding it a home within the administrative structure. The internal audit department, for example, will resist being given any line management tasks, while the programming staff dislike anyone whose rôle seems to be making their job more difficult.

Security teams thus tend to be 'reorganised' regularly, leading to a loss of continuity; a recent study shows, for example, that the average tenure of computer security managers at US government agencies is only seven months [H]. In the rare cases where a security department does manage to thrive, it usually has difficulties attracting and keeping good engineers, as they get bored once the initial development tasks have been completed.

These problems are not unknown to security equipment vendors, but they are more likely to flatter the customer and close the sale than to tell him that he needs help.

This leaves the company's managers as the only group with the motive to insist on good security. However, telling good security from bad is notoriously difficult, and many companies would admit that technical competence (of any kind) is hard to instil in managers, who fear that becoming specialised will sidetrack their careers.

Corporate politics can have an even worse effect, as we saw above: even where technical staff are aware of a security problem, they often keep quiet for fear of causing a powerful colleague to lose face.

Finally we come to the 'consultants': most banks buy their consultancy services from a small number of well known firms, and value an 'air of certainty and quality' over technical credentials. Many of these firms pretend to expertise which they do not possess, and cryptology is a field in which it is virtually impossible for an outsider to tell an expert from a charlatan. The author has seen a report on the security of a national ATM network switch, where the inspector (from an eminent firm of chartered accountants) completely failed to understand what encryption was, and under the heading of communications security remarked that the junction box was well enough locked up to keep vagrants out!

4.2 Confirmation of our analysis

It has recently become clear (despite the fog of official secrecy) that the military sector has suffered exactly the same kind of experiences that we described above. The most dramatic confirmation came at a workshop held in Cambridge in April 93 [M2], where a senior NSA scientist, having heard a talk by the author on some of these results, said that:

- the vast majority of security failures occur at the level of implementation detail
- the NSA is not cleverer than the civilian security community, just better informed of the threats. In particular, there are 'platoons' of people whose career speciality is studying and assessing threats of the kind discussed here
- the threat profiles developed by the NSA for its own use are classified

This was encouraging, as it shows that our work is both accurate and important. However, with hindsight, it could have been predicted. Kahn, for example, attributes the Russian disasters of World War 1 to the fact that their soldiers found the more sophisticated army cipher systems too hard to use, and reverted to using simple systems which the Germans could solve without great difficulty [K1].

More recently, Price's survey of US Department of Defence organisations has found that poor implementation is the main security problem there [P]: although a number of systems use 'trusted components', there are few, if any, operational systems which employ their features effectively. Indeed, it appears from his research that the availability of these components has had a negative effect, by fostering complacency: instead of working out a system's security requirements in a methodical way, designers just choose what they think is the appropriate security class of component and then regurgitate the description of this class as the security specification of the overall system.

The need for more emphasis on quality control is now gaining gradual acceptance in the military sector; the US Air Force, for example, is implementing the Japanese concept of 'total quality management' in its information security systems [SSWDC]. However, there is still a huge vested interest in the old way of doing things; many millions have been invested in TCSEC and ITSEC compliant products, and this investment is continuing. A more pragmatic approach, based on realistic appraisal of threats and of organisational and other human factors, will take a long time to become approved policy and universal practice.

Nonetheless both our work, and its military confirmation, indicate that a change in how we do cryptology and computer security is needed, and there are a number of signs that this change is starting to get under way.

5 A New Security Paradigm?

As more people become aware of the shortcomings of traditional approaches to computer security, the need for new paradigms gets raised from time to time. In fact, there are now workshops on the topic [NSP], and an increasing number of journal papers make some kind of reference to it.

It is clear from our work that, to be effective, this change must bring about a change of focus. Instead of worrying about what might possibly go wrong, we need to make a systematic study of what is likely to; and it seems that the core security business will shift from building and selling 'evaluated' products to an engineering discipline concerned with quality control processes within the client organisation.

When a paradigm shift occurs [K2], it is quite common for a research model to be imported from some other discipline in order to give structure to the newly emerging results. For example, Newton dressed up his dramatic results on mechanics in the clothing of Euclidean geometry, which gave them instant intellectual respectability; and although geometry was quickly superseded by calculus, it was a useful midwife at the birth of the new science. It also had a lasting influence in its emphasis on mathematical elegance and proof.

So one way for us to proceed would be to look around for alternative models which we might usefully import into the security domain. Here, it would seem that the relationship between secure systems and safety critical systems will be very important.

5.1 A new metaphor

Safety critical systems have been the subject of intensive study, and the field is in many ways more mature than computer security. There is also an interesting technical duality, in that while secure systems must do at most X, critical systems must do at least X; and while many secure systems must have the property that processes write up and read down, critical systems are the opposite in that they write down and read up. We might therefore expect that many of the concepts would go across, and again it is the US Air Force which has discovered this to be the case [JAJP]. The relationship between security and safety has also been investigated by other researchers [BMD].

There is no room here for a treatise on software engineering for safety critical systems, of which there are a number of introductory articles available [C4]. We will mention only four very basic points [M3]:

1. The specification should list all possible failure modes of the system. This should include every substantially new accident or incident which has ever been reported and which is relevant to the equipment being specified.
2. The specification should make clear what strategy has been adopted to prevent each of these failure modes, or at least make them acceptably unlikely.
3. The specification should then explain in detail how each of these failure management strategies is implemented, including the consequences when each single component, subroutine or subassembly of the system itself fails. This explanation must be assessed by independent experts, and it must cover not just technical design factors, but training and operational issues too. If the procedure when an engine fails is to fly on with the other engine, then what skills does a pilot need to do this, and what are the procedures whereby these skills are acquired, kept current and tested?
4. The certification program must test whether the equipment can in fact be operated by people with the level of skill and experience assumed in the specification. It must also include a monitoring program whereby all incidents are reported to both the equipment manufacturer and the certification body.

These points tie in exactly with our findings (and with the NSA's stated experience). However, even a cursory comparison with the ITSEC programme shows that this has a long way to go. As we mentioned in the introduction, noone seems so far to have attempted even the first stage of the safety engineering process for commercial cryptographic systems.

As for the other three stages, it is clear that ITSEC (and TCSEC) will have to change radically. Component-oriented security standards and architectures tend to ignore the two most important factors, which are the system aspect and the human element; in particular, they fail to ensure that the skills and performance required of various kinds of staff are included, together with the hardware and software, in the certification loop.

5.2 The competing philosophies

Within the field of critical systems, there are a number of competing approaches. The first is epitomised by railway signalling systems, and seeks either to provide multiple redundant interlocks or to base the safety features on the integrity of a kernel of hardware and software which can be subjected to formal verification [CW].

The second is the aviation paradigm which we introduced at the beginning of this article; here the quality engineering process is based on constant top level feedback and incremental improvement. This feedback also occurs at lower levels, with various distinct subsystems (pilot training, maintenance, airworthiness certification, traffic control, navigational aids, ...) interacting in fairly well understood ways with each other.

Of these two models, the first is more reductionist and the second more holist. They are not mutually exclusive (formal verification of avionics is not a bad thing, unless people then start to trust it too much); the main difference is one of system philosophy.

The most basic aspect of this is that in signalling systems, the system is in control; if the train driver falls asleep, or goes through a red light, the train will stop automatically. His task has been progressively deskilled until his main function is to see that the train stops precisely at the platform (and in some modern railways, even this task is performed automatically, with the result that driverless trains are beginning to enter service).

In civil aviation, on the other hand, the pilot remains firmly in command, and progress has made his job ever more complex and demanding. It was recently revealed, for example, that Boeing 747 autopilots have for 22 years been subject to erratic failures, which can result in the plane starting to roll.

Boeing's response was blunt: autopilots 'are designed to assist and supplement the pilot's capabilities and not replace them', the company said [CR]. 'This means our airplanes are designed so pilots are the final control authority and it means that a well trained crew is the first line of safety.'

5.3 The computer security implications

Both the railway and airline models find reflections in current security practice and research. The former model is dominant, due to the TCSEC/ITSEC emphasis on kernelisation and formal methods. In addition to the conventional multilevel secure evaluated products, kernelisation has been used at the application layer as well [A2] [C5].

Nonetheless, we must consider whether this is the right paradigm to adopt. Do we wish to make the computer security officer's job even more mechanical, and perhaps automate it entirely? This is the direction in which current trends seem to lead, and if our parallel with signalling systems is accurate, it is probably a blind alley; we should follow the aviation paradigm instead.

Another analogy is presented in [BGS], where it is argued that the traditional centralised model of security is like the old communist approach to economic management, and suffers from the same limitations. The authors there argue that to cope with a world of heterogeneous networks in which no single security policy is able to predominate, we need an infrastructure which enables information owners to control and trade their own property, rather than trusting everything to a centralised administrative structure.

This analogy from economics would, if developed, lead to somewhat similar conclusions to those which we draw from comparing railway signals with air traffic control systems. No doubt many other analogies will be explored over the next few years; the key point seems to be that, to be useful, a security metaphor should address not just the technical issues, but the organisational ones as well.

6 Conclusions

Designers of cryptographic systems have suffered from a lack of information about how their products fail in practice, as opposed to how they might fail in theory. This lack of feedback has led to a false threat model being accepted. Designers focussed on what could possibly go wrong, rather than on what was likely to; and many of their products are so complex and tricky to use that they are rarely used properly.

As a result, most security failures are due to implementation and management errors. One specific consequence has been a spate of ATM fraud, which has not just caused financial losses, but has also caused at least one miscarriage of justice and has eroded confidence in the UK banking system. There has also been a military cost; the details remain classified, but its existence has at last been admitted.

Our work also shows that component-level certification, as embodied in both the ITSEC and TCSEC programs, is unlikely to achieve its stated goals. This, too, has been admitted indirectly by the military (at least in the USA); and we would recommend that the next versions of these standards take much more account of the environments in which the components are to be used, and especially the system and human factors.

Most interesting of all, however, is the lesson that the bulk of computer security research and development activity is expended on activities which are of marginal relevance to real needs. A paradigm shift is underway, and a number of recent threads point towards a fusion of security with software engineering, or at the very least to an influx of software engineering ideas.

Our work also raises some very basic questions about goals, and about how the psychology of a design interacts with organisational structure. Should we aim to automate the security process, or enable it to be managed? Do we control or facilitate? Should we aim for monolithic systems, or devise strategies to cope with diversity? Either way, the tools and the concepts are becoming available. At least we should be aware that we have the choice.

Acknowledgement: I owe a significant debt to Karen Sparck Jones, who went through the manuscript of this paper and ruthlessly struck out all the jargon. Without her help, it would have been readable only by specialists.

References

[A1] D Austin, "Marking the Cards", in *Banking Technology*, Dec 91/Jan 92, pp 18 - 21

[A2] RJ Anderson, "UEPS - A Second Generation Electronic Wallet". in *Computer Security - ESORICS 92*, Springer LNCS **648**, pp 411 - 418

[B] M Buckler MP, letter to plaintiff's solicitor, 8 June 1992

[BAB] "Card Fraud: Banking's Boom Sector", in *Banking Automation Bulletin for Europe*, Mar 92, pp 1 - 5

[BAN] M Burrows, M Abadi and RM Needham, *'A Logic of Authentication'*, DEC SRC Research Report **39**

[BB] "Cash Dispenser Security", *Barclays Briefing* (press release) 12/9/92

[BGS] JA Bull, L Gong, K Sollins, "Towards Security in an Open Systems Federation", in *Proceedings of ESORICS 92*, Springer LNCS **648** pp 3 - 20

[BMD] A Burns, JA McDermid, JE Dobson, *'On the meaning of safety and security'*, University of Newcastle upon Tyne Computer Laboratory **TR 382** (5/92)

[C1] A Collins, "Bank worker guilty of ATM fraud", in *Sunday Times*, 22 Mar 1992

[C2] A Collins, "The Machines That Never Go Wrong", in *Computer Weekly*, 27 June 1992, pp 24 - 25

[C3] D Coppersmith, "The Data Encryption Standard (DES) and its strength against attacks", IBM Thomas J Watson Research Center technical report **RC 18613 (81421)**, 22 December 1992

[C4] J Cullyer, "Safety-critical systems", in *Computing and Control Engineering Journal* **2** no 5 (Sep 91) pp 202 - 210

[C5] B Christianson, "Document Integrity in CSCW", in *Proc. Cambridge Workshop on Formal Methods* (1993, to appear)

[CR] Boeing News Digest, quoted in usenet newsgroup 'comp.risks' **14** no 5 (29 April 1993)

[CW] J Cullyer, W Wong, "Application of formal methods to railway signalling - a case study", in *Computing and Control Engineering Journal* **4** no 1 (Feb 93) pp 15 - 22

[DP] DW Davies and WL Price, *‘Security for Computer Networks’*, John Wiley and Sons 1984.

[E] J Essinger, *‘ATM Networks - Their Organisation, Security and Future’*, Elsevier 1987

[GO] G Garon and R Outerbridge, “DES Watch: An examination of the Sufficiency of the Data Encryption Standard for Financial Institution Information Security in the 1990’s, in *Cryptologia*, **XV**, no. 3 (July 1991) pp 177 - 193

[H] HJ Highland, “Perspectives in Information Technology Security”, in *Proceedings of the 1992 IFIP Congress, ‘Education and Society’, IFIP A-13 vol II* (1992) pp 440 - 446

[ITSEC] *‘Information Technology Security Evaluation Criteria’*, June 1991, EC document COM(90) 314

[J1] RB Jack (chairman), *‘Banking services: law and practice report by the Review Committee’*, HMSO, London, 1989

[J2] K Johnson, “One Less Thing to Believe In: Fraud at Fake Cash Machine”, in *New York Times* 13 May 1993 p 1

[JAJP] HL Johnson, C Arvin, E Jenkinson, R Pierce, “Integrity and assurance of service protection in a large, multipurpose, critical system” in *proceedings of the 15th National Computer Security Conference*, NIST (1992) pp 252 - 261

[JC] Dorothy Judd v Citibank, *435 NYS, 2d series, pp 210 - 212, 107 Misc.2d 526*

[JDKLM] DB Johnson, GM Dolan, MJ Kelly, AV Le, SM Matyas, “Common Cryptographic Architecture Application Programming Interface”, in *IBM Systems Journal* **30** no 2 (1991) pp 130 - 150

[K1] D Kahn, *‘The Codebreakers’*, Macmillan 1967

[K2] TS Kuhn, *‘The Structure of Scientific Revolutions’*, Chicago 1970

[L1] B Lewis, “How to rob a bank the cashcard way”, in *Sunday Telegraph* 25th April 1992 p 5

[L2] D Lane, “Where Cash is King”, in *Banking Technology*, October 1992, pp 38 - 41

[M1] S McConnell, “Barclays defends its cash machines”, in *The Times*, 7 November 1992

[M2] R Morris, invited lecture given at Cambridge 1993 formal methods workshop (proceedings to appear)

[M3] JA McDermid, “Issues in the Development of Safety Critical Systems”, *public lecture, 3rd February 1993*

[MB] McConville & others v Barclays Bank & others, High Court of Justice Queen’s Bench Division 1992 ORB no.812

[MBW] McConville & others v Barclays Bank & others *cit*, affidavit by D Whalley

[MM] CH Meyer and SM Matyas, *‘Cryptography: A New Dimension in Computer Data Security’*, John Wiley and Sons 1982.

[N] I Newton, ‘Philosophiae Naturalis Principia Mathematica’, University of California Press 1973

[NSM] *‘Network security Module - Application Developer’s Manual’*, Computer Security Associates, 1990

[NSP] *New Security Paradigms Workshop*, 2-5 August 1993, proceedings to be published by the ACM.

[P] WR Price, “Issues to Consider When Using Evaluated Products to Implement Secure Mission Systems”, in *Proceedings of the 15th National Computer Security Conference*, National Institute of Standards and Technology (1992) pp 292 - 299

[R] RA Rueppel, “Criticism of ISO CD 11166 Banking: Key Management by Means of Asymmetric Algorithms”, in *Proceedings of 3rd Symposium of State and Progress of Research in Cryptography*, Fondazione Ugo Bordoni, Rome 1993

[RM] R v Moon, *Hastings Crown Court*, Feb 92

[RSH] R v Stone and Hider, *Winchester Crown Court* July 1991

[S] A Stone, “ATM cards & fraud”, *manuscript 1993*

[SSWDC] L Sutterfield, T Schell, G White, K Doster and D Cuiskelly, “A Model for the Measurement of Computer Security Posture”, in *Proceedings of the 15th National Computer Security Conference*, NIST (1992) pp 379 - 388

[TCSEC] *‘Trusted Computer System Evaluation Criteria*, US Department of Defense, 5200.28-STD, December 1985

[VSM] *‘VISA Security Module Operations Manual’*, VISA, 1986

[W1] G Welchman, *The Hut Six Story*, McGraw-Hill, 1982

[W2] MA Wright, ‘Security Controls in ATM Systems’, in *Computer Fraud and Security Bulletin*, November 1991, pp 11 - 14

[W3] K Wong, ‘Data security - watch out for the new computer criminals”, in *Computer Fraud and Security Bulletin*, April 1987, pp 7 - 13