Making Smartcard Systems Robust

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

Abstract

Smartcards are often sold as the solution to almost all information security problems. However, placing too much faith in any technology can lead to credibility problems; recent ATM disputes in Norway - and the problems experienced by the pay-TV industry - have shown that some smartcard systems do fail. We discuss their failure modes, and show how a prudent designer can minimise the risk of failure by making his system robust. We explore the nature of robust security: it affects all levels of a system, from security goals through the functionality and protocol levels to the cryptographic algorithms and their interactions. Finally, we discuss a fielded payment system, UEPS, which may provide a paradigm case of how to build a robust smartcard application.

1 Introduction

One of the smartcard vendors' most powerful arguments is the ease with which magnetic strip cards can be forged. Forgery has not only caused direct losses to card issuers and customers, but has led in some countries to serious embarrassment: highly publicised disputes between banks and their customers erode confidence and cast doubt on the integrity of the whole payments system.

For example, phantom withdrawals from automatic teller machines (ATMs) have been the main source of complaints to the UK financial sector ombudsmen in recent years. Yet despite a report from a parliamentary commission of enquiry into banking law which concluded that the system of personal identification numbers was insecure [J], the ombudsmen have followed the banks' line that their systems are infallible and refused to award compensation [A1].

Some of the resulting court cases have been very controversial. In one recent trial, a policeman who had complained about six phantom withdrawals from his account was accused of attempting to obtain money by deception and convicted. The resulting press outcry [E] unnerved the banking industry, and the institution responsible publicly denied that it asked for a prosecution (contrary to the evidence given by their fraud manager in court).

Bad publicity is the banker's nightmare, and preventing it striking at payment systems is a goal of all central banks. However, there is nothing magic about smartcards which prevents application designers making blunders which break their systems' security. This has happened with with satellite TV decoder systems and with prepayment electricity meters [AB]; it now appears to have struck the financial sector too, with disputes in Norway which may be a harbinger of problems to come.

2 Credibility Problems in Norway

In Norway, as in a number of other countries, the banks invested millions in issuing their customers with smartcards, and are now quite adamant (at least in public) that no debit can possibly be made to a customer account without the actual card and PIN issued to the customer. Yet a number of thefts at the University of Trondheim have cast serious doubt on their position.

In these cases, smartcards were stolen from offices on campus and used in ATMs and shops in the town; the victims, who include highly credible witnesses such as senior academic staff and overseas visitors, are quite certain that their PINs could not have been compromised. In any case, the nearest ATM is several kilometres away, so if the victims were observed while using their cards, then the thief must have followed them a considerable distance.

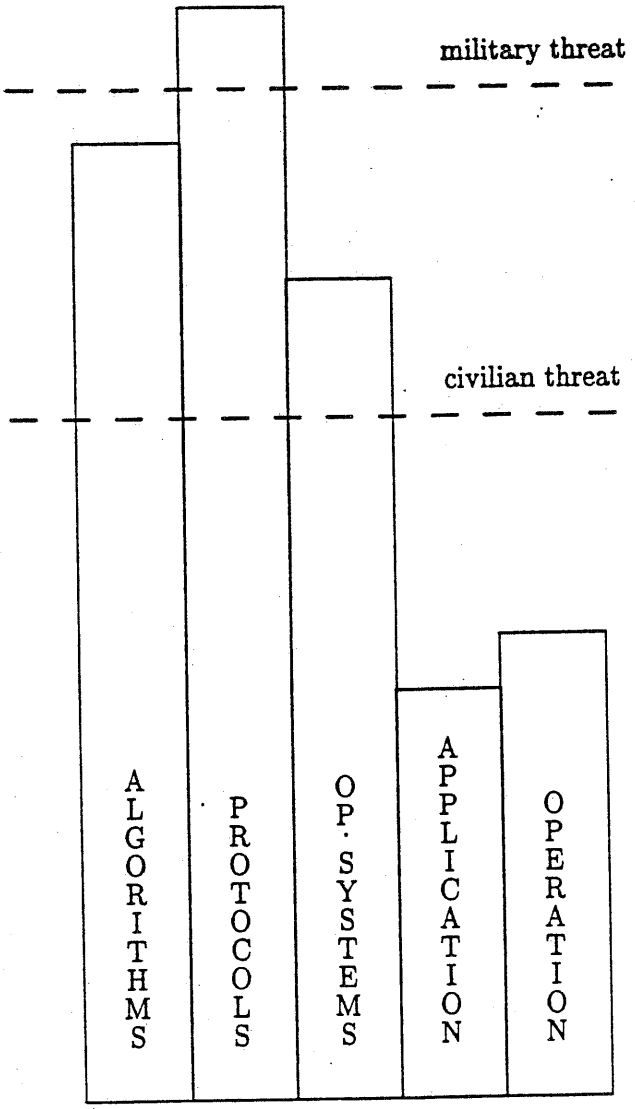
The Norwegian banking ombudsman (Bankklagenemnda), when faced with a complaint from one of the victims, asked the banks for technical details on the system in use. The banking association (Bankforeningen) refused, claiming that *'an important part of the security in the card systems is that the routines are not documented and publicly known. Such an independent investigation will therefore reduce the level of security. Bankforeningen is therefore against letting for instance a consultancy company acquire knowledge of the routines in order to investigate them'* [BN].

The ombudsman then went to Norges Bank, the country's central bank, and asked it to investigate the system. Norges Bank agreed, but even it was not provided with any technical information; Bankforeningen simply gave it 'managerial' assurances that the system was sound, which were passed on to the ombudsman. He then used these assurances an an excuse to deny compensation to the victim.

By now, the reader may be starting to wince, and will wince more when told that the disputed transactions violated the card cycle limits: although only NOK5000 should have been available from ATMs and NOK6000 from eftpos, the thief managed somehow to withdraw NOK18000. The extra NOK7000 was refunded without any explanation.

3 The Nature of Civilian and Military Threats

Our studies of security failure in magnetic strip payment systems [A1] [A2] showed that the vast majority of security failures which led to actual losses occurred at the level of implementation or operation. The following diagram may give a useful conceptual model: the five columns show the amount of confidence available in cryptographic algorithms, protocols, operating systems, applications and operational controls respectively, while the two horizontal lines show the 'military' and 'civilian' threat levels - a capable motivated opponent and an opportunistic opponent respectively.



Algorithms We have a number of encryption algorithms which are thought to be good (DES, IDEA, ...), but since certain government agencies knew about differential cryptanalysis twenty years ago [C], there remains the fear that one of these agencies might find a shortcut attack on our algorithm. But while this may be a serious concern for military system builders, it need not concern most commercial designers overmuch.

Protocols Thanks to formal methods such as the BAN logic [BAN], and to robustness properties which will be described shortly, the protocol security problem can be considered solved. This does not of course mean that there are no weak protocols around; on the contrary, as the successes of the BAN authors [AN] and the recent controversy over ISO 11166 [R] [A3] showed, and recent attacks on satellite TV scrambling systems have confirmed, there are plenty fielded systems which are insecure. However, a diligent designer can now avoid such attacks completely by using publicly available techniques.

Operating systems: Operating systems have also been extensively studied, and (thanks to TCSEC and ITSEC) there are a number of products on the market, ranging from commercial discretionary access control products to military multilevel systems. Of course, military products are usually too expensive and out of date to be used in commerce and industry; and weaknesses are still found in military products from time to time. However, experience shows that if one disregards viruses and worms, the features of the operating system are almost never the main cause of real attacks on commercial systems.

Applications: Here at last we come into the world of real commercial threats. A large number of attacks on electronic payment systems have been due to application programming blunders [A2], and it seems highly likely that some such error is also responsible for the Norwegian problem. In military systems, too, there may be attacks on cryptographic algorithms, but there are many more attacks which result from application blunders [M1].

Operations: Operational blunders also cause many security failures; no matter how good the security technology, sloppy management can render it useless. For example, in August 1993, my wife went into a branch of a major English bank, and told them that she had forgotten her PIN; they helpfully printed a replacement PIN mailer from a PC behind the counter. This was not the branch at which our account is kept; no-one knew her, and the only 'identification' she produced was her bank card and our cheque book. By that time, banks in Britain had endured some eighteen months of bad publicity about poor ATM security (with which my name had been associated), and this particular bank had been a press target since April of that year. With management this dreadful, it makes no difference whether the card itself is magnetic or silicon based.

The moral is that if a customer is to get anything like full value from an investment in smartcard technology - which will usually cost many millions - then the application will have to be designed, implemented and managed in a much more robust way than has often been the case in the past. Now that VISA and MasterCard have agreed to move their enormous customer base from magnetic strip cards to chip cards over the next six years, we are about to see an investment whose sheer size may freeze certain aspects of the technology for many years at the level achieved by about 1996. If the robustness aspects are not got right, and the architectures adopted by banking networks are fragile, then the industry's credibility could be imperilled. Robustness should therefore be an urgent concern of smartcard vendors.

4 The Nature of Robustness

The big question is therefore: how do we build robust applications using smartcards? Without this, the product adds little value. However, there has until now been no generally accepted idea of what robust security consists of.

In the rest of engineering, the exact nature of robustness varies from one discipline to another. Bridge builders know that most possible faults, such as poor steel, contaminated concrete, or uneven weathering, just reduce the structure's breaking strain slightly; so the usual rule is to design a bridge so that its theoretical breaking strain with optimal materials is six times what is required, and to proof test samples of the actual construction materials to half that, or three times the needed strength [P].

In aircraft engineering, on the other hand, many accidents are caused by the failure of critical components, and so designers make extensive use of redundancy. With very critical functions, this may extend to design diversity: for example, a typical modern airliner will determine its flight attitude using two electronic flight instrumentation systems, which use gyros driven by the main power circuits; but if these both fail for some reason, there is a 1950's technology artificial horizon with pneumatic gyros, and a 1920's vintage turn and slip indicator driven by a battery.

But neither overdesign nor redundancy is enough for the secure systems designer. Just using a number of weak encryption algorithms one after another will not necessarily have the same effect as using a strong algorithm; and the unthinking use of redundancy in computer systems can be dangerous, as resilience can mask faults which would otherwise by found and fixed.

However, the fact that blunders are responsible for most real security failures in both civilian and military computer systems shows that robustness is just as much a problem as it is for people building bridges or aircraft. We might expect it to be even harder to pin down to a general principle, since we have to consider what it means at a number of levels. We will look at the algorithm, protocol, application and management levels in turn.

4.1 Robustness of Algorithms and their Interactions

As we noted above, cryptographic algorithms are not the most pressing concern of the commercial secure systems designer. We visit this topic briefly because our work on algorithm interaction gave the insight that algorithm robustness is about explicitness.

Various authors had looked at the problem of how we are to prevent unwanted interaction between cryptographic algorithms, such as between a hash function and a digital signature algorithm. Since the Diffie Hellman paper in 1976 [DH], it had been recognised that hash functions should be one-way, in that it is easy to work out $h(M)$ from M but hard to find a suitable M given only $h(M)$. This was difficult to formalise, because of the implicit temporal ordering, so a second property was proposed, of collision freedom: that it is hard to find two different inputs M and M' such that $h(M) = h(M')$ [D].

For a number of years, cryptographers were content to stipulate that hash function primitives should be one-way and collision free. Then in 1992, Okamoto proposed a third primitive, correlation freedom: a function h is correlation free

if it is hard to find two different inputs M and M' such that $h(M)$ and $h(M')$ differ in only a few bits. He conjectured that correlation freedom was a strictly stronger property than collision freedom.

Last year, we showed that this conjecture was true. In fact, we showed that there are many properties which we might want a hash function to have, such as multiplication freedom (we can't find X, Y and Z such that $h(X)h(Y) = h(Z)$) and addition freedom (ditto $h(X) + h(Y) = h(Z)$), and we showed that these properties were independent of each other by constructing functions which had some freedom properties but not others [A4]. This gave us the insight that, at the application level at least, we have to be explicit about the properties which our application requires from any cryptographic primitives we use.

4.2 Robustness of Protocols and Operating Systems

Robustness in cryptographic protocols made its formal début in the academic literature in May 1993: three papers appeared in which it was proposed as a solution to the problem of designing authentication schemes.

1. At the 1993 Cambridge Workshop, Li Gong argued that most protocol errors occur when people try to be smart, and called for a protocol equivalent of structured programming. For example, by putting in the name of the sender and the recipient of each message in a protocol, and by insisting on freshness, most of the cut-and-paste attacks in the literature could be prevented [G].

2. A month later, at Eurocrypt 93, Boyd and Mao suggested that a protocol should be called robust if authenticating any message depends only on information contained in the message itself or already in the possession of the recipient, and that the purpose of a message in a robust protocol should be capable of being deduced without knowledge of the context [BM].

3. The following day, at Oakland 93, Woo and Lam proposed that protocols be made robust by chaining successive steps, and developed a formal semantics for protocols which are immune to cut-and-paste attacks (in their terminology, such protocols possess the 'correspondence' property). They argued that these are absolutely fundamental to proving other security properties [WL1].

Each of these proposals, which appeared logically simultaneously, tackles part of the problem. However, none of them is adequate on its own; protocol failures are known which result from the lack of name, freshness and context information within the security envelope. The interested reader is referred to [AN] for examples.

Our approach to the problem is that explicitness is the proper organising principle for security robustness. This is put forward in [A2], and draws on two

sources - firstly, the work on algorithm interaction mentioned in the previous section, and secondly, the experience of a system which used robust protocols and which we implemented in 1991 [A5]. We will discuss this system, UEPS, in section 5 below.

Explicitness subsumes the above three proposals - lack of name, lack of freshness and lack of context are all failures of explicitness. In each case, some information - which both parties know implicitly - is not explicitly expressed within the security envelope, and can thus be manipulated by an opponent.

Explicit protocols have been criticised as expensive [S]; and Woo and Lam have recently experimented with starting from an explicit protocol and looking to see what information can be safely taken out [WL2]. However, the philosophical point here is that optimisation is a process of replacing something which works with something which almost works, but is cheaper; and if the 'optimised' version is not cheaper at all, then the process is undesirable. In fact, the implicit information in a cryptographic protocol (such as 'this is message 3 in a run of Kerberos version 6 subprotocol 17') is already known to both parties, so there is no added communication cost; and as most real protocols either use fast symmetric ciphers, or use fast hash functions to reduce messages prior to calculating a digital signature, it is quite unclear that there is ever a significant computational cost either.

In fact, the protocols used in UEPS [A5] are efficient as well as robust. When designing them, we bundled in everything for two reasons - firstly, to save on code space and to avoid expanding the communications protocol (we had limited ROM and very slow offline devices for inter-card transactions), and secondly, to facilitate the formal verification of the protocol.

4.3 Application Program Robustness

Much of the robustness needed in application programs will be a matter for software engineering. However, even here explicitness plays a significant rôle, with many common methodologies emphasising the need to elicit requirements in the most explicit possible form, to validate these by test cases or prototyping, and to ensure that they are coded and tested properly. However, there are some aspects of the application layer which deserve special attention.

Many older banking systems have the problem that application programmers have discretion over what security features to implement. Consider for example the security modules used in networks of ATMs to encrypt personal identification numbers [VSM]. These devices return a whole series of response codes which may be significant to the security manager: for example, if a programmer starts making unauthorised experiments with live keys, he will probably give rise to a key parity error message.

It would thus clearly be prudent to monitor these response codes, and the obvious solution is to write a device driver which intercepts all abnormal response codes from the security module and brings them to the attention of the

audit or security staff. However, no institution we have seen implements proper monitoring; as the system will 'work' without it, and as device drivers are tricky to write, the job never gets done.

One of advantage of a trusted computing base (TCB), whether built from smartcards or from security modules, is that we can design the TCB's transaction set in such a way as that application programmers have to do all the necessary checks. This was done in UEPS by chaining each message to the next; thus a programmer who tries to cut corners by discarding some message information will find that he cannot make the system work at all.

Of course, this strategy can be more productive with a distributed TCB, such as we can implement with smartcards, as the checking can be made much more pervasive. We can manage security state information, such as response codes and certain aspects of transaction history, more intelligently where there is a component of the TCB in every logical node of the network, rather than just having one security processor attached to a mainframe which runs perhaps a hundred security critical processes such as branch accounting, interbank reconciliation and audit.

Suppose for example that we allow bank branch staff to initialise cards for customers. In countries where poor telecommunications force us to work largely offline we will typically have no choice but to do this; UEPS, for example, uses a teller card to load some crypto keys and other initialisation data to the customer card. Suppose we also expect that about 1% of our branch staff will try to embezzle money in an average year [A1]; we might then decide to put the issuing teller's identity, not just on each card, but in every transaction as well. In this way, we can provide a distributed audit system, and open new possibilities for fraud detection.

The key point is that smartcards are objects which can retain security state; and this state need not be limited to the customer's account balance. A well designed system will use it to provide much higher assurance of application functionality than was previously available, and to ensure that partial, insecure implementations do not work at all.

4.4 Robustness at the Management Level: Making Goals Explicit

As noted above, the major management mistake made by banks in the UK and elsewhere was that they did not make their security goals explicit, and in particular they did not recognise the need for a fair arbitration procedure. To quote Tom Watson's 1977 Oxford address:

> 'One of the most important problems we face today, as techniques and systems become more and more pervasive, is the risk of missing that fine, human point that may well make the difference between success and failure, fair and unfair, right and wrong ... no IBM computer has an education in the humanities' [W]

Quite simply, failures are inevitable, and if there is no mechanism to deal with them, then the bank will have to choose between paying all claims and paying none of them. The courts have forced US banks to take the former course [JC], which leaves them reliant on ATM cameras to prevent malicious claims; while banks in the UK and Norway opted for the latter - at the cost of bad publicity, litigation and loss of public confidence.

Furthermore, when banks claim (as the Norwegians did) that their security systems cannot withstand public scrutiny, this also shows that their security goals were not properly thought through. Disclosure of evidence is one of the most ingrained features of the legal systems in free societies - an accused person has the right to see and to challenge every link in the chain of evidence against him, and in most countries the requirements for discovery in civil cases are hardly less strict. Every cryptographic system whose main purpose is to establish liability must be built on the assumption that it will have to withstand scrutiny by hostile expert witnesses in court [A6].

The danger of missing the fine human points seems to be especially acute when designing smartcard based systems, as the technology seems impressive, and the temptation to hubris is correspondingly strong.

4.5 Tying it All Together: Relating Goals to Mechanisms

Finally, we need to be very careful when following a line of reasoning from goals - however well researched, human friendly, legal and thoroughly agreed - to mechanisms. This part of the security design process is well known to be fraught with danger.

Morris reports an interesting explicitness requirement imposed by the NSA in its own internal evaluations [M2]. Each product must have not just an operations manual describing its normal use, but also an attack manual which describes every combination of technical attacks, blunders, and subversion of personnel which could succeed in defeating it. The attack manual may be highly classified and available only to a small number of evaluators, but it must exist, and it must be comprehensive; if the evaluators can find any other significant attacks, the product will be rejected.

This innovation may be valuable in the commercial world as well, and in general, the explicitness principle can be propagated down through the design in a structured way, using techniques developed by the safety critical systems community [A1].

However, where the principal function of a cryptographic system is establishing liability, then the real test must be whether it can stand up in court against hostile expert witnesses. Here, one needs standards of best industry practice, as courts will use these to determine which party to a dispute has been more negligent [A6]. For this reason, we will now discuss some of the robustness features of UEPS.

5 Robust Payment Systems: Best Industry Practice

UEPS, the Universal Electronic Payment System, was designed during 1991 by systems house Net One for its client the Permanent Building Society in Johannesburg, and implemented later that year; its purpose was to extend modern banking services at low cost to South Africa's black population.

The Permanent Building Society, which merged with Nedbank to become NedPerm, had more black customers than any other financial institution in South Africa; but most of these customers only had savings book accounts, and the charges traditionally made for cheque clearance (up to R12 or about $4) were so high relative to black incomes as to prevent the spread of cheque accounts to this sector of the market.

It was felt that a new product delivery system was needed, which should cater for point-of-sale transactions and keep costs as low as possible. However, the country's poor telecommunications meant that many transactions would have to be carried out offline, and even if the telephone lines had been available, online processing imposes fairly high costs on point-of-sale systems as very high availability is required. Thus offline processing was chosen, and this prevented the use of traditional magnetic stripe cards, as the forgery risk would have been unacceptable. Finally, high levels of illiteracy suggested the use of PINs rather than signatures to authorise transactions.

These requirements drove the design of UEPS, which has been a commercial success. After a trial from 1991 - 1993 under NedPerm's brand name 'Megalink', during which there was no single case of fraud detected, it was adopted by all four major South African banking groups with only minor changes. It is now projected that there will be 2 million cards in issue by early 1995.

The system has been sold to other customers too. It is being introduced by banks in the neighbouring country of Namibia, and a version of it is used by South African Breweries to manage accounts with tavern and liquor store owners. Most recently, after a year's trial, it has been adopted by Sperbank, a large savings bank in Russia, which plans to deploy it during February/March 1995 at 35 branches in 14 regions of the former Soviet Union.

The system relies on value transfer between smartcards. A customer loads her card with money from a card held by a bank teller or installed in an ATM; she then makes purchases by transferring value to a merchant card; and the merchant in turn uploads his takings to his bank via an ATM or terminal.

5.1 Algorithm robustness

We chose DES as the encryption algorithm as the client was comfortable with it and as public key smartcards were not available in 1991. Because keysearch was already considered a threat by the banking community [GO], the only robustness feature incorporated at the algorithm level was double encryption (this is admittedly one feature which is more overdesign than explicitness!).

5.2 Protocol robustness

The payment protocols used in UEPS have the robustness property that all the mutual information between the two parties is made explicit, by being incorporated into the message keys. A run of a protocol between cards A and B, using key pair K1 and K2, and message blocks A1, B1, A2, ... looks like this:

A $\longrightarrow$ B: K1(K2(A, B, A1))

B $\longrightarrow$ A: K1(K3(B, A, B1)) where K3 = K2(A, B, A1)

A $\longrightarrow$ B: K1(K4(A, B, A2)) where K4 = K3(B, A, B1)

The primary design objectives here were to implement both message chaining and double encryption using the minimum possible amount of code space, and to make formal verification of the protocols easier (for details of the verification see [A5]).

Since the Megalink prototype, key diversification has been introduced. The original system had the same keypair K1 and K2 present in every card (albeit one was loaded at the factory and the other at initialisation); in the new system, only the cards embedded in bank and merchant terminals possess a set of universal secrets, and the customer cards have keys derived from their serial numbers using these master keys.

5.3 Goal and Application Robustness - Resolution of Disputes

We did not repeat the ATM designers' mistake of forgetting to provide a means of arbitrating disputed transactions. In fact, a number of features were built into UEPS explicitly for dispute resolution.

Firstly, there are two signatures on each digital payment: one generated with a key known only to the issuing bank and the customer card, and one generated with a key controlled by the clearing house and loaded by them to the card before it is supplied to the bank. The latter signature is checked before funds are credited to the retailer presenting the cheque, while the former would only be checked in the event of a dispute. Having separate signatures was considered important given that the primary legal liability is from the card issuing bank to its customer, and that an unknown number of banks of varying levels of technical sophistication were expected to join the scheme.

A further feature, not mentioned in [A5], was writing encrypted audit trails on the transaction receipt using two separate keys, one known to the customer and his card issuing bank but not the merchant or the clearing house, and the other known to the merchant and the clearing house but not the customer or his bank. In the event of a dispute getting as far as trial, courts are likely to accept transaction logs as evidence, and especially logs which are not written on magnetic media but on paper - and of which copies are kept by both the customer and the retailer. This solves most of the problems with computer evidence discussed in [A6].

Incidentally, although the key used by the customer's card to encrypt a receipt record can be recreated by his bank's security module, and the merchant's key by the central clearing house, this need not have been the case; the keys could exist nowhere else except in the card which uses them. In that case, one would require any customer (or merchant) who disputes a transaction to produce his card to an arbitrator in order to decrypt the record, This shows that, even where we are using a symmetric algorithm, cryptographic keys do not have to be shared to provide a valuable service. This is counter to the general intuition, and relies ultimately on the assumption that the card is tamper evident.

Another benefit of having a number of independent mechanisms is that in the (hopefully rare) event of a technical attack, it is unlikely that all the mechanisms will be broken. After all, some of the mechanisms (such as the encrypted audit records) do not have to be circumvented in order to obtain value from the system, and in fact could not be circumvented unless the attacker had access to the victim's card (or to the bank's security module).

It follows that, with a high degree of probability, the existence of an attack would become evident. This avoids the risk of placing an unmeetable burden of proof on the complainant, which was what decided the US courts against the banking industry in magnetic strip card disputes [JC].

Finally, there is a rule that all merchant transactions must be banked within 14 days. This means that, unlike the similar Mondex system now proposed in the UK [MA], UEPS can refund lost customer cards after a short delay. In any case, settlement functions are a central part of traditional banking systems; abandoning them, as seems implicit in many electronic cash designs, could give rise to unpredictable risks.

6 Conclusion

In the real world, most things break sooner or later. Traditional engineers usually tackle this problem by making systems robust, and this may involve some combination of overdesign and redundancy; however, we have shown that robust computer security depends on explicitness as much as anything else.

This must be used in a structured way, and the starting point is to make the business goals explicit. Next we need to establish an accurate threat model and work though to the functional properties which the system needs. Security is not after all a simple Boolean attribute, but a set of properties which enable a system to fulfil its purpose (of processing banking transactions, helping to win wars, or whatever).

Once we have an explicit set of security properties which our system must have, we can look for ways in which we can use our trusted computing base to enforce them. We can also use application layer features to enforce good operational practice, and thus ensure that all our functional properties interlock

strongly. The explicitness principle is also a useful guide to designing the technical features, such as cryptographic protocols, on which the whole structure is built.

Finally, robustness as explicitness is not new; we built it into the design of UEPS, a commercially successful banking system, back in 1991. That exercise showed that there are no significant computational or communications costs associated with robust protocols. The only real investment involved in building a robust security system is some extra effort at the design stage; but doing things right the first time is always cheaper in the long run.

References

[A1] RJ Anderson, "Why Cryptosystems Fail", in *Proceedings of the 1993 ACM Conference on Computer and Communications Security* pp 215 - 227

[A2] RJ Anderson, "Why Cryptosystems Fail", to appear in *Communications of the ACM, November 1994*

[A3] RJ Anderson, "A Note on ISO 11166", presented at the Crypto 94 rump session

[A4] RJ Anderson, "The Classification of Hash Functions", in *Proceedings of the 4th IMA Conference in Cryptography and Coding (1993)* (proceedings to be published by OUP)

[A5] RJ Anderson, "UEPS - A Second Generation Electronic Wallet", in *Computer Security - ESORICS 92*, Springer LNCS v 648 pp 411 - 418

[A6] RJ Anderson, "Liability and Computer Security - Nine Principles", in *Computer Security - ESORICS 94*, Springer LNCS (to appear)

[AB] RJ Anderson, SJ Bezuidenhout, "On the Security of Prepayment Metering Systems", *to appear*

[AN] M Abadí, RM Needham, 'Prudent Engineering Practice for Cryptographic Protocols', in *Proceedings of the 1994 IEEE Symposium on Security and privacy* (to appear)

[BAN] M Burrows, M Abadí and RM Needham, 'A logic of Authentication', *Report 39, Digital Systems Research Center, Palo Alto, Ca.*

[BM] C Boyd, WB Mao, 'Limitations of Logical Analysis of Cryptographic Protocols', in *Pre-Proceedings of Eurocrypt 93* pp T88 - T96

[BN] Behne v Den Norske Bank, Bankklageneminda, Sak nr: 92457/93111

[C] 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

[D] IB Damgård, "Collision free hash functions and public key signature schemes", in *Advances in Cryptology - EUROCRYPT 87*, Springer LNCS v 304, pp 203 - 216

[DH] W Diffie and ME Hellman, "New Directions in Cryptography", in *IEEE Transactions on Information Theory*, v IT-22 no 6 (November 1976) p 650

[E] B Ellis, 'Prosecuted for complaint over cash machine', in *The Sunday Times*, 27th March 1994, section 5 page 1

[ECMA] European Computer Manufacturers' Association, *"Secure Information Processing versus the Concept of Product Evaluation"*, technical report 64 (December 1993)

[G] L Gong, 'Thoughts on Cryptographic Protocols', in *Proceedings of the 1993 Cambridge Protocols Workshop*, Springer LNCS (to appear)

[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

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

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

[M1] R Morris, in *Proceedings of the 1993 Cambridge Protocols Workshop, Springer LNCS (to appear)*

[M2] R Morris, in *Proceedings of the 1994 Cambridge Protocols Workshop, Springer LNCS (to appear)*

[MA] H McKenzie, D Austin, 'Banks ready to do business with smart cards', in *Banking Technology v 11 no 1* (Feb 94) p 4

[O] T Okamoto, "Provably Secure and Practical Identification Schemes and Corresponding Signature Schemes", in *Abstracts of Crypto 92*, pp 1-15 to 1-25

bibitem[R]R RA Rueppel, "Criticism of the ISO CD 11166 banking-key management by means of asymmetric algorithms", in *Proc. 3rd Sym. on State and Progress of Research in Cryptography* pp 191 - 198

[S] PF Syverson, 'Adding Time to a Logic of Authentication', in *Proceedings of the 1993 ACM Conference on Computer and Communications Security* pp 97 - 101

[ST] "How £200 can buy account details", in *Sunday Times* 29 November 1992 p 1 and p 3

[VSM] *"VISA Security Module Operations Manual"*, VISA 1986

[W] 'Former IBM Chief TJ Watson Jr dies', in *IEEE Computer v 27 no 2* (Feb 94) p 84

[WL1] TYC Woo, SS Lam, 'A Semantic Model for Authentication Protocols', in *Proceedings of the 1993 IEEE Computer Society Symposium on Research in Security and Privacy* pp 178 - 194

[WL2] TYC Woo, SS Lam, "A Lesson on Authentication Protocol Design", in *Operating Systems Review* v 28 no 3 (July 94) pp 24 - 37