Chapter 10 ELECTRONIC FUNDS TRANSFER AND THE INTELLIGENT TOKEN

10.1. INTRODUCTION

Payments can be made by cash, cheque, credit card and in many other ways. New electronic methods of payment are being introduced to improve the speed and convenience and to reduce costs by eliminating paper vouchers. In addition to these objectives there is the paramount requirement of security against attempts to defraud banks or their customers. The possibility of electronic payment systems depends on the relationship of trust between banks and their customers.

Each payment begins with an instruction by the payer. It must be possible to identify the source of the instruction so that the authority to make the payment can be checked. Traditionally, payments between bank accounts have been initiated by a signed instruction from the customer to his bank, such as a cheque, but there are now other forms of payment using paper documents such as credit-card vouchers and credit transfers. Since the essence of these documents is in their information, they can be replaced by digital messages if the necessary checking of authority is possible. The taking of cash from a cash dispenser, using a plastic card and a personal number, illustrates how far we have come from the concept of a signed authority for each payment, yet this is a type of transaction that has not had serious security problems. On the other hand, credit cards, depending for their authority on a signature, have been the target for fraud on an increasing scale. Each payment mechanism brings with it new security questions.

Electronic funds transfer (EFT) operates in 'retail banking' to give the customer a more convenient service, like 24-h service from cash dispensers, to save expense by avoiding the need for capturing data manually from paper vouchers and to improve security as much as possible. When EFT operates between banks and from corporate customers to banks, it serves mainly for convenience and speed of transaction.

Experience shows clearly that, with careful design, EFT can be much better protected against fraud than any method which uses paper documents. New technology such as public key ciphers and digital signatures may be able to offer even greater protection. In this chapter we shall describe some representative EFT systems, giving an indication of their security requirements and the available technology to meet them. We will begin with existing and well-understood systems and move towards those that are still in planning or no more than a future possibility.

When cash dispensers and automatic teller machines (ATM) were introduced,

the need to protect them against fraud was obvious and, with a few exceptions, these systems have been reasonably secure. Growing evidence of fraud in other systems with poor protection has made it clear that any new payment method should be very secure in its protection against fraud. No system can be completely secure, so the design should be matched to an assumed level of technology in the way the system may be attacked, with a margin of safety against attacks at a higher level still.

The need for greater care in the design of automated systems is not just a reaction to the growing sophistication of fraudsters. When the protection against fraud is based on human vigilance, the enemy cannot be sure how this is being applied and what rules are in operation at a given time. New precautions can be added where they are needed without upsetting the whole system. In this respect, automated systems are less flexible and the security measures have to be designed and built into the system from the start. These precautions can be studied by the enemy over a long period and an elaborate attack can be prepared if the potential gains are large enough. Because the system is automated there is a danger that the fraud may also be automated in the sense that, once a successful attack has been devised, it can be repeated rapidly to produce a substantial gain to the enemy (and loss to the bank or its customers). Furthermore, the essence of the new systems is their convenience to the user, so any bank which had to withdraw a payment system because of mounting fraud would lose many customers. These are reasons why the design of future electronic banking systems requires a much greater attention to data security than was ever the case in the past, and larger safety margins.

The security of major systems is not preserved primarily by secrecy in their basic design. Discussion of the principles of security in EFT is valuable because it underlines the risks and, in the long run, leads to better methods. But it would be wrong to describe in detail the ways in which working systems can be attacked. In this chapter we shall describe the principles governing the design of various payment mechanisms without describing details which are too close to the potential vulnerabilities of any actual system.

From a cryptographic viewpoint, the attacks on payment systems are active attacks which seek to change data for the benefit of the attacker. Therefore the precautions are primarily those of *authentication* and *integrity* which enable unauthorized changes to be detected, so that payment transactions which are fraudulent can be inhibited before any damage is done. It seems that the need for data security has been understood in banking earlier than in other activities where data security may, in fact, be an equally urgent requirement. Banking provides an excellent example for the application of principles described in the rest of this book. The techniques used in this chapter come from earlier parts of the book, and here it is their application which interests us. We rely heavily on the verification of personal identity, the subject of chapter 7, but the predominant means of identification of payments has been the password. In the form used in banking, the password is usually short because the customer is expected to remember it and it is known as a *personal identification number* with the acronym PIN. When the password includes letters as well as numerals it is sometimes known as a *personal identification code* or PIC, but we will use the name PIN for either kind.

To begin this chapter there is a review of payment methods. Then three general types of payment are introduced and studied from a security viewpoint. We begin with payments between banks where, because of the degree of trust, the security requirements are more easily met, but the risks are great because of the high value of payments. We continue with payments between banks and their customers, such as cash dispenser transactions.

Throughout this chapter we shall be describing payment mechanisms that are implemented by many different kinds of financial instruction such as banks, savings and loans and credit-card issuers. To avoid the expression 'financial institution' at every point we shall refer to them conventionally as 'banks'. When their role in a transaction is specific, they will be denoted by that role, for example 'card issuer'.

An increased complexity appears in the security measures for ATMs when these are shared between a number of banks. The problems are similar to those of 'point-of-sale' payment systems which are the subject of the next section and both have the characteristic that the banks which might lose money from security violations are remote from the point of transaction and dependent on telecommunications. Thus shared ATMs and point-of-sale payment systems usually employ messages between central processors and remote terminals. The possibility of off-line systems for point-of-sale payments is discussed, and this gives a possible role for *intelligent tokens*, sometimes called 'smart cards'.

Having discussed payments from bank to bank and transactions between customer and bank, the final section of this chapter looks at payments between customers in which the bank need not be on-line at the time of payment. These methods depend essentially on the public key signature and there is no current proposal for their widespread introduction, but they are worth studying as a future possibility. They also point to the future value of digital signatures in almost all kinds of payment system.

10.2. ESTABLISHED PAYMENT MECHANISMS

The three main payment systems in operation today are cash, cheques and credit cards. Cash payments are the greatest in number and cheque payments the greatest in value. For example in USA, in the mid 1970s, cash payments formed about 88 per cent of the 300 billion transactions per year.[1] (We use the convention that 1 billion = 10^9.) Of the $7 000 billion transacted in these payments approximately 96 per cent was by cheque even though the survey excluded payments over $10 000. There is some uncertainty about the cost per transaction of these payment methods, but it is clear that cash payments were the least expensive at about 1.5 cents per transaction whereas cheques and credit-card payments each cost about 50 cents at that time. The cost of cash payment lies in the manufacture and replacement of coins and notes and there is a difficult trade-off between the cost of manufacturing notes and their resistance to counterfeiting. Like all anti-forgery measures, this resolves into a technical battle, in this case between the forgers and the legitimate bank-note printers and paper makers.

The bank cheque

The cheque payment mechanism is based on a signed instruction from an account holder to his bank to make the payment. This instruction or cheque is given to the payee as evidence of the payment so that, if the payer can be trusted, the payment can be regarded as complete, enabling the goods or services to be supplied. Figure 10.1 shows the outline of this payment mechanism which is surprisingly complicated and at first sight not a good candidate for automation. The payee or beneficiary, Bill, presents the cheque for payment to his own bank which sends the cheque to the payer's bank since the instructions contained in the cheque are destined for them. The payer's bank debits Ann's account and the payee's bank credits Bill's. The settlement is made between banks in the form of a payment covering all the transactions in a given period, such as those in one day.

Since there are many banks, bilateral arrangements between banks would be clumsy, so both the handling of the cheques and the inter-bank payment is usually provided by a 'clearing system'. As the volume of cheques increases (at about 7 per cent per year) an efficient clearing system has become essential. The big practical disadvantage of the cheque system is its slowness. Even with automation, several days must elapse before Bill's bank can be sure that the cheque has been cleared and allow Bill to use the funds that have been paid to him. The period of uncertainty prevents either Ann or Bill from using the funds for several days. Where very large payments are concerned, this is a serious cost to the customers and it favours the banks, which can use the amounts in transit to gain interest.

The physical sorting of cheques to return them to their payers' banks once threatened to undermine the cheque payment system and it was saved only by the introduction of cheque sorting using magnetic ink characters. In principle, the whole system could work without paper as soon as the essential data have been captured, the cheque itself remaining at the branch of Bill's bank where it was presented for payment. The use of 'cheque truncation' depends on the legal system under which the banks are operating.

The security of the cheque mechanism depends on the manual signature on

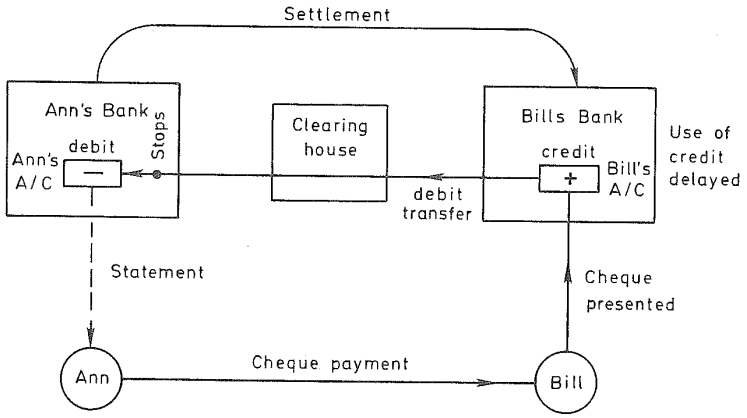

Fig. 10.1 Cheque payment mechanism

the cheque. The weakness of the signature as a security measure is that the enormous volume of cheques which are handled prevents the verification of individual signatures in most cases. Dynamic signature methods which were described in chapter 7 are of no value here, but the automatic verification of signature patterns on paper (the static signature) is showing some promise. Nevertheless, cheque fraud is a considerable problem. Cheque guarantee cards give some protection to shop-keepers, up to a limit set by the banks, in return for checking the signature and copying a number on to the cheque from the cheque guarantee card, but the cost of fraud then falls on the banks.

The paper cheque with its manual signature is not an ideal subject for automation yet it embodies an important principle, which is the authority given by the payer and made apparent to the beneficiary before it threads its way through the clearing system. Changed into a modern form as a message with a digital signature this can be regarded as an 'electronic cheque'. At the end of this chapter we show that this kind of message can be a basis of many different payment systems.

Credit transfer

The cheque is a *debit transfer* when it passes from Bill's bank to Ann's bank because it asks the latter to debit Ann's account. Other transactions are *credit transfers* because they carry the payment with them, for example in the payments between banks which make the settlements for cheques. When an account holder at a bank makes a 'standing order' for regular payments, these payments are carried out by credit transfers. For example, a magnetic tape from the bank of payer Ann may contain among its many transactions a request to credit Bill with a stated sum. The magnetic tape containing this payment also entitles Bill's bank to debit the account it holds for Ann's bank so that the total of the transactions on the tape cancel out. The transport of these magnetic tapes can be replaced by file transfers using data communication.

Figure 10.2 shows the effect of a typical credit transfer from Ann's account in

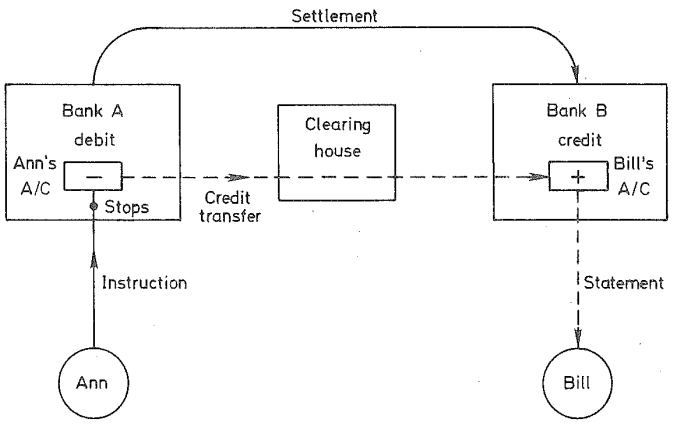

Fig. 10.2 Credit transfer

bank A to Bill's account in bank B. If it starts with a paper document, this can end its journey at A, while the information flows on. The settlement between the banks covers a multitude of individual transactions. Sometimes the credit transfers are initiated by a magnetic tape containing a large number of transactions. In the UK, a bank customer can make an arrangement to send the information directly to the 'Bankers' Automated Clearing Service' where the sorting of entries takes place. For this purpose, tapes or diskettes carry the transactions, or they are sent over a public data network.

A form of banking, named a Giro, which favours credit transfers, is common in European countries. This is a single 'bank' which can very easily make credit transfers between accounts held by its own customers — the two banks and the clearing house of Figure 10.2 are merged into one organization. Because of the need to let Bill know he has been paid, a short statement of account is made whenever a credit arrives and (since the paper is all in one place) the credit transfer form itself goes with that statement as confirmation of the payment and its source.

The Giro has a long history in Europe where Post Offices operate the systems using the postal service to carry the paper work. Typically, Ann sends a credit transfer to the Giro centre by post, which then makes the transaction by adjusting the accounts of both Ann and Bill, then sends an advice to Bill that the payment has been made. Since the two accounts are adjusted simultaneously, there is no 'float' on which the Giro obtains interest. The first notice to Bill of the completion of the payment comes with the advice, so the completion of the deal in goods or services may have to wait on the two transits through the postal system, and posts are not getting any faster. The Postal Giro has an extra convenience that the document can also contain the order for goods or services, making one letter complete the deal. Since Bill receives the payment with the order, there must be some trust, so this simple and convenient device is used for small payments. Giro payments and credit transfers generally are particularly useful for regular payments that are not constant in value such as for telephone service, utilities, tax and insurance.

The simplicity of the postal Giro is due to the monopoly position of national Giro services, which means that the whole transaction between customers' accounts takes place in one centre. There is no problem of paper-sorting or settlement between clearing banks. It follows that Giro services can be automated easily, except for the postal part. Eventually the cost of the post will make this less economic than on-line electronic payments. Bank credit transfers enable many payments to be covered by one cheque and the credit transfers, once their data have been captured, are easily automated because the instruction on paper stays at the customer's own bank. At present, the credit transfer seems better able to fit in with information processing than the debit transfer or cheque, but if paper documents do not have to be moved around (for example if they stay at the point-of-sale in an on-line system) the distinction is less clear. The route taken by messages in point-of-sale EFT is, in fact, closer to that of the cheque.

Summary of the properties of payment methods

Returning to the three principal methods of payment used today, their advantages and disadvantages can be summarized. Cash is the cheapest to operate and

loses very little by fraud. The payment is quick and there is no 'clearing delay'. It is not surprising that cash is the most frequent payment method. The main objection is that cash can be stolen. Its disadvantages are found in collecting large quantities of small payments (public transport, payphones and parking meters) and in large payments when violent crime is a danger. Cheques are best for large amounts when the cheque can be cleared in time or the payer can be trusted (or brought to court successfully). For small payments in shops they are too easily used for fraud, too expensive and payment is slow. For very large payments, the clearing delay loses interest to the customers and favours the bank. Credit transfers are best for multiple and large payments. Credit cards have characteristics similar to cheques, except those concerning payment delay, where the customer has more choice. Considering their disadvantages, cheques and credit cards should be overtaken by more automated payment mechanisms as soon as the economics of on-line operation are favourable and the security of electronic methods is assured. Cash will remain the most frequent payment method.

10.3. INTER-BANK PAYMENTS

A large customer of a bank may present a magnetic tape containing thousands of credit transfers. In the UK these can be handled directly by Bankers Automated Clearing Services (BACS) where the items are sorted according to the destination banks and passed on to them also in batch form on magnetic tape. In the future, most of these batches will be handled by file transfer through a data network. The security requirements are principally the authentication of the files to prevent unauthorized changes and it was for this purpose that MAA was designed. Electronic funds transfer based on the batch handling of credit transactions has been in operation for a long time. This service is provided by BACS for large- and medium-sized bank customers and is the world's largest clearing house in terms of transaction volume.

When very large sums are paid between bank customers a cheque is not used because it puts the funds out of use for too long. In its place, telephone or Telex messages can be sent by one bank to give an immediate request to another bank to make a payment on its behalf. The payer makes the request to his bank. That bank validates the request, making sure that it is genuine and that the individual has sufficient funds, then passes on the request to the beneficiary's bank which makes the payment, debiting the account which the payer's bank holds with it. In this way the payment is completed within minutes or hours instead of days. Because telephone or Telex messages are vulnerable to fraud by impersonation it has become normal to add to each message an authenticator using secret keys established between correspondent banks for this purpose. Traditionally, the name given to this authenticator is a 'test key'. The secret keys employed were often in a form of tables, typically two sets of tables held by different individuals. The calculation of test keys was necessarily rather simple because it was carried out by a clerk.

The volume of these payments increased and so did their complexity. Errors due to misunderstanding of the complex transactions had to be resolved by telephone calls and their resolution depended on a degree of trust betwen correspondent banks. These semi-automatic payments continue but are increasingly being replaced by use of custom-built, message-handling systems of which the

prime example is the International Society for Worldwide Interbank Financial Telecommunications (S.W.I.F.T.) system. Following on the introduction of S.W.I.F.T. national systems have been introduced such as clearing houses inter-bank payment system (CHIPS) in USA and clearing houses automated payments system (CHAPS) in UK. After describing the S.W.I.F.T. system, we shall compare its service, and the method of providing it, with CHAPS.

The Society for Worldwide Inter-bank Financial Telecommunication S.C.

The society is a bank-owned, non-profit cooperative society, directed by more than 1400 shareholding member banks which are located in more than 60 member countries. It began as a result of a study by European banks in the late 1960s which led to the formation of the company in 1973 and the first operation of its network in 1977. It has members in North, Central and South America, Europe, Africa, Australasia and the Far East. The costs of its message transfer service are paid for by members using a tariff designed to recover the costs according to numbers of connections, addresses and volume of message traffic. In addition to providing the service and the associated information and training it acts as the forum for agreement on standards.

All messages are handled by store and forward procedures through one or both of the 'operating centres' located in Belgium and USA. The society installs, in member countries, its 'regional processors' which are connected by leased lines to the operating centres with some extra connections so that single line failures do not disconnect any country. Countries with heavy traffic such as USA, UK, Germany and Italy have multiple regional processors and each regional processor has duplicate CPUs for reliability. All connections to the network go through the regional processors which act as concentrators. The operating centres contain duplicate equipment and customers can fall back to an alternative regional processor so that the worst effect of an equipment failure is a short delay. Figure 10.3 shows part of the network as it was in 1984. Since then, the communications system has been redesigned for SWIFTII.

A S.W.I.F.T. terminal is generally a small computer which can 'stand alone' or be used as a front end to the bank's payment system computer. The majority of connections use a 'S.W.I.F.T. Interface Device' (SID) with software supported by S.W.I.F.T. but about 30 per cent use other methods, principally the 'Direct S.W.I.F.T. Link' (DSL) software running on International Business Machines (IBM) main frames. For small users, and as a fall back for others, Telex was used but this has not proved convenient and a microprocessor-based terminal offered by a S.W.I.F.T. subsidiary company has taken over this function.

The S.W.I.F.T. system can report the number of messages sent and received and stores the messages it has transmitted so that they can be retrieved until 14 days have elapsed. This helps banks to resolve any difficulties about the content of messages or the completeness of their traffic.

Message format standards

Strict standards for message format can be imposed because the composition of messages is assisted by the SID in nearly all cases. The agreement on standards has been one of the major advances brought by the S.W.I.F.T. system because

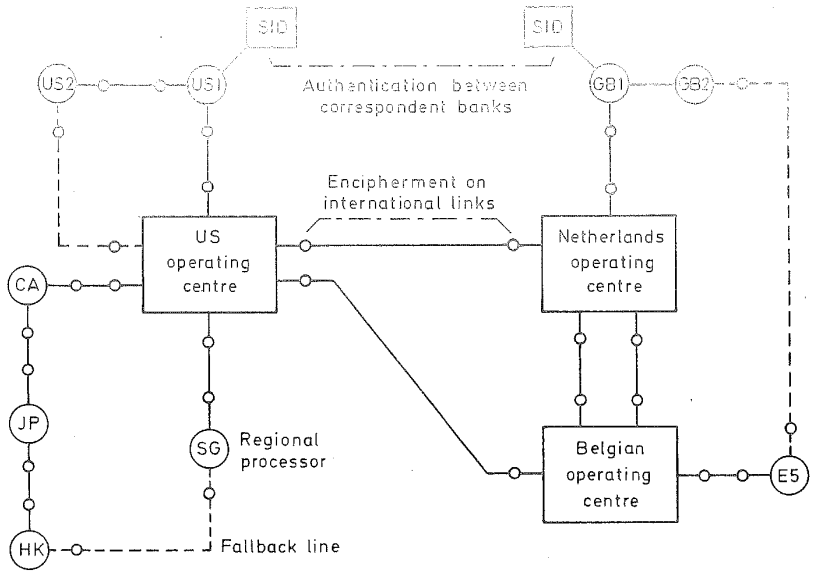

Fig. 10.3 Part of the S.W.I.F.T. network (1984)

its uniform terminology and notation prevent some of the misunderstandings that might otherwise occur in complex interactions. The nature of this complexity will appear later. Working groups of S.W.I.F.T. members derive these standards by discussion and agreement and continually review and update the standards for new applications of the system.

Each message format is denoted by a message type (MT), for example MT 100 is the type known as *customer transfer* which is an inter-bank payment order originated by a bank's customer in favour of another bank's customer. Message types are divided into categories according to their first digit and there are at present eight categories (number 6 is not defined) which are listed in Figure 10.4. In each category n, the groups numberd n90 to n99 are for purposes common to all categories, for example group MT 499 is a free format message concerning documentary collection. The names of categories are self evident except perhaps for categories 4 and 7.

'Documentary collection' is a service provided to an exporter of goods to enable him to receive payment at the place of delivery in return for the shipping documents which give title to the goods. The exporter's bank instructs a bank at the place of delivery, which collects the payment for him. This avoids payment in advance, where the buyer is exposed, or billing the buyer after delivery (open account) where the exporter is exposed. An alternative way to handle the situation is 'documentary credit', for which category 7 messages are designed. The buyer instructs his bank which provides credit (up to a certain sum and time limit) enabling a bank in touch with the exporter to make payment on receipt of stipulated documents. The buyer receives the documents in return for the money, as before, the difference being that the exporter has already been paid.

A message contains a number of fields some of which are mandatory and others

Category	Example groups
1. Customer transfers	100 Customer transfer
2. Bank transfers	202 Bank transfer in favour of third bank
3. Foreign exchange, loans/deposits	300 Foreign exchange confirmation 350 Advice of loan/deposit interest payment
4. Documentary collections	400 Advice of payment
5. Securities	50n Orders and offers
6. Reserved for future use	
7. Documentary credits	71n Issue and amendment
8. Special payment mechanisms	88n Bank card authorization
9. Special messages	900 Confirmation of debit

Some of the message groups have standards that are agreed but are not yet implemented.

Others exist for interim use while awaiting finalization of standards.

Fig. 10.4 Examples of S.W.I.F.T. message groups and categories

optional. The demarcation of each field is by a message tag, for example the tag '15:' denotes the test key. Taking as an example the customer transfer (MT 100), the mandatory fields are

20: transaction reference number
32A: value date, currency code, amount
50: ordering customer
59: beneficiary customer

The fields are sufficient for the simplest kind of transaction where the ordering customer — the person making the payment — has an account with the bank which sends the message and the beneficiary customer has an account with the bank which receives the message. The sending and receiving bank do not appear as fields in the message text because they are contained in the header of the message which directs it through the system.

For technical and accounting reasons, not all pairs of banks can usefully exchange messages. Full connectivity would imply a million or more possible links each with an authentication key and account relationship, which is not practical. Therefore, smaller banks specialize in transfers with certain countries, offering this service in competition with other banks. Consequentrly the accounts of the ordering customer and the beneficiary customer may not be in the banks sending and receiving the messages.

As many as four other banks in addition to the sending and receiving bank may be involved in the transaction and it is this kind of complexity which led to errors when payment messages were less formal, before S.W.I.F.T. existed. Figure 10.5 shows the messages and payment in the most complex case. These messages have fields denoting

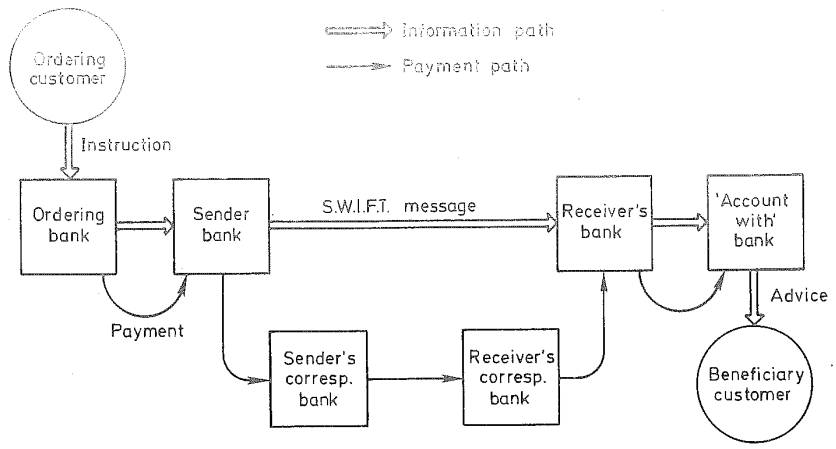

Fig. 10.5 A complex customer transfer by S.W.I.F.T.

52: ordering bank
53: sender's correspondent bank
54: receiver's correspondent bank
57: 'account with' bank.

The ordering bank sends a message to the sending bank (which originates the S.W.I.F.T. message) and also transmits the funds to this bank. The receiving bank makes the payment to the beneficiary via the bank which holds his account, that is the 'account with' bank.

The payment between sending and receiving bank is often made by an account held by one of these banks for the other. Whether this is the sender's account with the receiver bank or the receiver's account with the sender bank depends on the currency in which the transaction is made. If the currency is that of the country of the receiver bank then the receiver debits an account held by the sender in this bank. Alternatively, if the currency is that of the country of the sending bank, the sending bank credits the account of the receiver which it holds for him. Credits and debits for these accounts and statements of account can be sent as special messages in category 9.

In some cases it is necessary for the payment to go through intermediate banks working either on behalf of the sender or receiver. The message says that the payment will be made this way by quoting the correspondent banks in the fields with tags 53: or 54:. Any of the additional four banks referred to in these optional fields can be present or absent in a customer transfer message. Messages to the sender's and receiver's correspondent banks to effect these payments are quite independent of the S.W.I.F.T. message which goes direct from sending to receiving bank.

Formats are defined in a similar way for a large number of message types. New message types are being developed. In each message group there is the possibility to send free format messages for purposes not covered in the standards,

using the digits 99 to signify free format, for example 199 in the customer-transfer group.

There is no doubt that the careful definition of these message types and their formats has been an important byproduct of the introduction of the S.W.I.F.T. message service.

Security in the S.W.I.F.T. system

The most important security features are authenticity and integrity. There have been a number of spectacular frauds by generating bogus payment messages between banks under manual systems of operation. In each case the authentication of messages was at fault.

The general properties of an authenticator algorithm were described in chapter 5 with some examples. The details of the S.W.I.F.T. authenticator are not public knowledge. As in any authenticator, the sharing of this key between sender and receiver bank and its secrecy from all others is the basic requirement for secure authentication.

The responsibilities for authentication are clearly defined. The algorithm is provided by S.W.I.F.T. (and in some cases the software in a terminal within which the algorithm is implemented). Keys are exchanged bilaterally between banks and are not known to S.W.I.F.T. or its personnel, but S.W.I.F.T. has issued guidelines about the exchange of authenticated keys suggesting the procedures to be used and the timing of key changes. Members can regard S.W.I.F.T. as a forum to assist their co-operation in the security matters but the responsibility for the safe exchange of keys lies entirely with the members.

An authenticator does not prevent messages being replicated, deleted or stored and retransmitted later. These requirements are handled by the sequence numbering of messages. Sequence numbering within the S.W.I.F.T. network ensures that messages are not lost or duplicated. The connection between S.W.I.F.T. and its users is safeguarded by input and output sequence numbers. The S.W.I.F.T. operating centre checks the format and input sequence number of messages offered to it and refuses those which have format errors or wrong sequence numbers. The output from the S.W.I.F.T. operating centre contains an output sequence number which must be checked by the receiver. The transaction reference number provides an end-to-end sequence control for each pair of banks and is included in the part of the message to which the authenticator is applied.

The interface between S.W.I.F.T. and a member bank must be protected against the introduction of false messages which have correct sequence numbers and authenticators. This is a matter for the individual banks who carry the ultimate responsibility for declaring that a message is valid and should be passed into the S.W.I.F.T. system. For this reason, SIDs and other connection arrangements normally provide for the assembly of a message by one person and the checking of the message and its release for transmission by a second person. The software and hardware of the connection must be protected against unauthorized changes because they administer these precautions and also store the authenticator algorithm and its keys.

Terminals coming on line to the system must verify their identity by a password. The passwords are issued by S.W.I.F.T. in the form of tables which are sent in

two parts, A and B. By sending them separately the interception of a complete password set is made less likely. Password tables must be acknowledged to S.W.I.F.T. by an authorized person before they are activated on the system. Each table contains a sequence of passwords listed against a sequence number. Each login employs the next password in the sequence so that interception of passwords by line tapping gives no clue to the next password that will be needed. To avoid the trick of extracting passwords by impersonating the S.W.I.F.T. system to the terminal, each password is given a response number which the user must receive from S.W.I.F.T. and check before beginning transactions. The formalities and procedures for introducing new password tables and the method of fall back when problems occur are carefully defined.

In the central part of the S.W.I.F.T. system which consists of the operating centres, regional processors and connecting lines, the security is the responsibility of the S.W.I.F.T. organization. Access to the system, its software and the users' messages is strictly controlled and so is physical access to the areas containing the computer systems. The international lines which connect operating centres together and join them to regional processors are protected by encryption to preserve the privacy of the banks' messages. It is a matter for the individual bank to decide whether to use encryption on the line between its SID and the regional processor but S.W.I.F.T. will offer help. Privacy, though important to some, is not such a cardinal security factor as authentication. This is why end-to-end encryption is not provided, whereas authentication is end-to-end. The authentication allows the use of keys which are not known to S.W.I.F.T. but end-to-end encryption would prevent the useful role of S.W.I.F.T. in storing, validating and retrieving messages.

The chief inspector's office has overall supervision of security and privacy and performs security audits. Internal and external auditing are carried out at random intervals to make sure that the security of data and of all S.W.I.F.T. operations is being maintained.

The S.W.I.F.T. system provides an excellent lesson in the careful design of system security. Similar systems like CHIPS and CHAPS, though they differ in detail, employ similar principles for authentication, sequence numbering, login and other security features.

The Clearing Houses Automated Payments System

The Clearing Houses Automated Payments System (CHAPS) network carries payment messages between banks, like S.W.I.F.T. The basic security requirements of the two systems are similar, but when they are examined in more detail, differences appear, due to their different institutional framework, geographical coverage and settlement methods. A comparison of the two systems is instructive.

S.W.I.F.T. provides the facility for carrying messages between any of its banks. Any pair of banks which use it make their own arrangements for keys to authenticate messages travelling between them and arrange a route for settlement of debts between them. CHAPS is operated by a small group of 'settlement banks', who all do substantial business with each other, so it is

a closer-knit community. These banks offer a means for large payments between about 300 other banks in the UK, most of them branches of foreign banks which use London as a business centre. The attraction of the settlement banks as intermediaries lies in their facility for *same-day* settlement using accounts they hold at the Bank of England. Cheque-clearing settlement in the UK takes place via CHAPS, in addition to its main function of payments between bank customers. The facility of same-day settlement for individual payments is called 'town clearing'. It has been done in the past by carrying paper documents within the compact business centre of 'the City' — a square mile (2.5 km^2) containing the offices of most of the 300 banks involved. In order to let the Bank of England arrange its funds before close of business, the individual payments stop at 15.00 hrs and the settlement between the thirteen banks follows immediately. About 19 000 payments took place per day with a total of £17 000 million. The movement of paper documents in the city is by bank messengers on foot.

The method worked well but had two limitations due to its old-established transport method. It was limited to the city. Banks elsewhere had to correspond with settlement banks by telephone or telex messages which was inconvenient and required careful discipline to keep it secure. The service was expensive per message and, though this cost was small compared with the interest saved by immediate access to funds, it led to a lower limit of £10 000 on individual transactions allowed to use the system. The CHAPS payment system is an automation of town clearing, using a computer-based message system between the settlement banks (of which the Bank of England is one). Some of these banks share gateways into the system, for economy. There is no limitation on where the point of entry to CHAPS is placed, which makes it possible for Scottish banks to have access points placed near to the banks they will serve. Additions to the group of settlement banks are not often made but further access points are feasible whenever they are needed.

The message format has been chosen with the aim of easy conversion to or from S.W.I.F.T. standards because instructions for payments via S.W.I.F.T. will often initiate payments through CHAPS. When a payment has been sent into CHAPS, settlement that day is part of the implied agreement, therefore a bank acting on a S.W.I.F.T. payment message must be sure of the source of the funds. Members of CHAPS compete in providing settlement facilities, balancing the value of the business against the risks they take. If a CHAPS payment has to be cancelled, this can only be done by a contrary payment, which needs the agreement of the payee.

Figure 10.6 shows the physical structure of CHAPS. Unlike S.W.I.F.T. it has no central operation, but uses messages transferred through the packet-switched public data network called Packet Switchstream (PSS). The integrity of the system depends on 'gateways' implemented in Tandem non-stop computers with software that has been jointly developed. Each message receives a logical acknowledgement via PSS back to the originating gateway, so that the performance of the system is continually monitored. The high availability of PSS and the Tandem computers is fundamental to the design. The gateways administer time stamps, sequence numbers and running totals for each bank pair. With each message from A to B, the total 'paid' by A to B since the last settlement is reported,

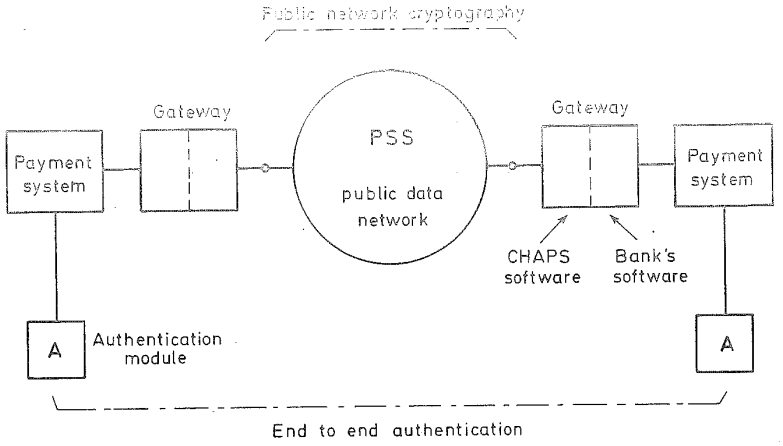

Fig. 10.6 Structure of the CHAPS system

like a bank statement that is always up-to-date. Consequently, at the end of business for the day, agreement on the accounts can be immediate and settlement follows by a message to the Bank of England.

Each bank's arrangements for message processing outside the CHAPS interface are its own concern. The large banks run a payment process in their own main-frames. Smaller banks (those with less CHAPS activity) can run their payment process in the Tandem that holds the gateway.

Clearing houses automated payments system is different from S.W.I.F.T. because same-day settlement through the central bank is an integral feature of its operation. The physical and organizational structure is also different. It is operated by a more centralized group of banks but the network has a more distributed structure. The security requirements are very similar to those of S.W.I.F.T. and are primarily concerned with authentication of messages and less critically with the confidentiality of messages.

The authenticator used in CHAPS is the Data Encryption Standard (DES) in its cipher-block chaining mode. It is implemented in the tamper-resistant hardware modules which hold the keys and perform the tasks of generating and checking authenticator values. The key management reflects the structure of CHAPS in which each pair of banks has an individual relationship. It employs master keys used to protect session keys which are transmitted over the network. Though the method of authentication has been agreed by the settlement banks, authentication is an end-to-end matter and is the responsibility of the two banks concerned. The payment system has been designed to be transparent to the message content, except for its procedure of handling the payment fields.

Confidentiality is provided by encipherment at the point of entry to PSS and decipherment at the exit from PSS. The main security feature is the authentication procedure, implemented in the bank's payment process with the aid of the special hardware module. Because the two processes of generating and checking authenticators are carried out in the physically secure modules, the ability to

generate authenticators can be confined to the sender. In this way, some of the useful properties of a signature are obtained in a system which uses a symmetric cipher, the DES.

10.4. AUTOMATIC TELLER MACHINES

Early automatic teller machines (ATMs) were essentially cash dispensers which had only one function, to deliver cash in the form of bank notes and debit a corresponding bank account. To identify the user, tokens were used in the form of cards of various kinds. Some early machines used punched cards that had only one life — they authorized one payment only and were not returned to the user, who had to obtain a supply of cards from his bank. There were also magnetic cards with a limited life, for example twenty withdrawals of cash. These early machines functioned for one bank, dispensing cash only to customers of that bank. To reduce the risk of lost cards being misused it was usual, right from the beginning of cash dispensers, to issue a personal identification number (PIN) to the customer which is entered on a keyboard to complete the identification. Many of these early cash dispensers were off-line; they worked on their own without connection to a central data base. They were located at first inside the bank where they provided an alternative to the human tellers. After some experience had been gained, cash dispensers were installed in the 'through the wall' position in which they were part of the bank building but available from outside it when the bank was closed. In this way banks could provide 24-h cash service — a factor which is becoming even more important with the growing competitiveness of banking organizations.

A third kind of location which was entirely away from the bank developed later. This can provide greater convenience but the cost of installation makes it suitable only in places with a large access to users, for example shopping centres and very large work areas which are remote from banks. The real value of these remotely sited cash dispensers or ATMs arrives when a single ATM can be used by the customers holding 'cash cards' of many different financial institutions.

The increase in remoteness from the bank increases the problems of physical security. One of the earliest crimes against 'through the wall' cash dispensers was to pull them completely out of the building with a heavy machine. Now these dispensers are built like a strong safe and fixed to the concrete foundation.

By adding further facilities, cash dispensers began to deserve the name of 'automatic tellers'. These can be used for enquiries about the status of accounts, for transfers between the customer's own accounts (such as deposit and checking accounts) and possibly for credit transfers from a customer's account to some other person's account. The machines can deliver travellers cheques instead of money and they can accept deposits, though in doing the latter they add little to the facilities of a bank deposit box. The most important function of automatic teller machines is still that of dispensing cash (or travellers cheques in countries where these are needed) and this is the operation requiring the greatest attention to security because any weakness which allowed cash to be withdrawn without authority would quickly render the service untenable to the banks. One of the primary requirements of a cash dispenser or ATM transaction is, therefore, to identify the customer as reliably as possible.

Automatic teller machines raised a host of legal problems concerning the liabilities of the bank and the customers when things go wrong. In the USA this has been covered by legislation, both state and federal. In other countries banks have been able to make their own agreements with customers and there has been comparatively little problem with disputes.

Generally we can say that the ATM facilities are greatly liked by bank customers, to the extent that short queues form at ATMs even when human tellers are free. Where banking hours are short, the use of ATMs outside hours has enabled banks to compete with other financial institutions having longer opening hours and these have, in turn, been persuaded to install ATMs to give 24-h service.

On-line and off-line operation

There are several advantages in connecting an ATM to a central data base so that transactions can be checked before money is paid out. The account balance can be checked and the card identity compared with a central file of cards against which payments will not be made — the so-called 'hot card list', 'negative file' or 'blacklist'. But on-line connection is expensive and has in the past been unreliable in some countries because of communication problems. So there is a strong incentive to use off-line operation if the security risks are not too high.

Checking the PIN against the card identity can be done in the terminal if the PIN is related to the account number and other card data by an algorithm, for example the use of a cipher with a secret key. Some details of these methods are described later. Having the relationship dependent on a secret key which is the same for all ATMs is a risk, because the key is held at so many places. Compromise of the key makes the whole system unsafe because an enemy is able to discover the PIN of any stolen cards or make his own cards and determine the PIN which will match them. The physical security surrounding an ATM makes it very unlikely that the key will be discovered by intrusion. The weakness, if any, is in the way the key is loaded into the ATM. The key can be the result of a calculation employing data from several sources or the algorithm can be a complex one requiring several keys. It can then be arranged that no one person has access to all the data necessary to forge cards.

The keys are used in two places in the system, at the ATMs for checking the relationship of the PIN with the account number and other card data and at the bank headquarters where PINs are calculated for distribution to customers. It is probably in the central area that the security of the keys is most at risk and this type of risk appears in some form in an on-line system, whatever method is used centrally for checking PINs. Therefore the additional risk in an off-line ATM system due to the multiplicity of places at which PIN checking is carried out can be small if the system is designed well. The physical security of the ATMs is an essential factor in this evaluation.

An off-line terminal can be equipped with a blacklist and this can be updated periodically. Since the bank's liability typically begins when the stolen card is reported, there is a strong incentive to update the blacklist quickly, within a day at most. The organization needed to do this increases the effective cost of off-line systems.

The greatest danger in off-line systems lies in the lack of effective communication between one ATM and another through the central data base. For example, if a card is stolen and the PIN is known or can be discovered and if the enemy can make several copies of this card, these can be used rapidly in many ATMs and an off-line system will not react by blacklisting the card until the next updating cycle. In an on-line system, the unusual activity on a particular account is easily detected and stopped. An on-line system can enforce complex rules limiting the usage of a card such as: total amount withdrawn per day or week or month; or number of withdrawals per day. With an off-line system such rules are unenforceable except when the token itself can carry secure information about its usage.

In some early ATM systems, data about the card's usage were stored in an area of the magnetic stripe that was rewritten each time the card was used. The data were enciphered, but this protection is illusory because a card can be returned to an earlier state by restoring its data. An on-line system can protect against this 'replay' threat with a sequence number but the storage on the card is not needed. To make storage on the card effective it must be physically and logically protected. This is the 'intelligent token' which will be discussed later.

In an off-line system using magnetic-striped cards there is a great need to prevent the copying of cards or the manufacture of new ones. This is the purpose of a 'security feature' incorporated in the card. An example of a security feature is the EMI watermark (see page 179) which stores a number on the magnetic-striped card in a way which cannot be altered or forged. Someone who copies a card, even onto another card with magnetic watermark, will not be able to copy the watermark number. Because a mathematical relationship exists between the watermark data and other data on the card, any copied or forged card without this relationship can be detected. The simplest relationship to use is a cryptographic one with a secret key k. For example if W is the watermark number and I the other identifying information on the card, such as the account and card serial numbers, a reference number R is calculated where

$$R = \mathrm{E}k(I, W).$$

This is stored on the card. Only a card with the correct reference number will be accepted and the reference number cannot be calculated without a knowledge of k. The notation (I, W) indicates that the data in these fields are enciphered as a cryptographic chain, thus if either I or W is changed it is necessary to recompute R in order to generate a valid card. This, we assume, the enemy cannot do because the key k is stored securely. In essence, the reference number R is another kind of password.

It is possible to refine this method by making R a digital signature of the data contained in I and W. This signature is computed by the bank using the secret key when the card is generated but any enemy, using old or stolen cards (which have their own W values) to copy a card for which the PIN is known, will not be able to compute the new signature. The advantage is that a digital signature can be checked by using a public key, therefore the secret key k is used only centrally for card encoding. The disadvantage of this method is the large amount of data employed in a signature, typically 512 bits, which does not fit easily in typical magnetic-striped cards.

To summarize the comparison between off-line and on-line operation, there is no limit to the methods which can be applied to checking card validity in the on-line system but off-line systems are limited to checking the PIN against the card data, comparing with a local blacklist and at most preventing the repeated use of one card at a particular ATM. The security advantage lies very strongly with on-line operation. With the growing tendency towards shared ATM systems capable of carrying out transactions for many separate financial institutions, we shall find that there is an additional motive for on-line operation.

Some operators of ATMs require their terminals to continue giving service when the central data base is out of action, for example during routine maintenance. This involves an extra risk, which is reduced if the periods of off-line operation are not at regular times and are undetectable. But the off-line state of an ATM is revealed by its inability to give the account balance. The safest policy is to equip an ATM system in such a way that off-line periods are rarely necessary.

PIN management

Personal identification number (PIN) management is the subject of a US National Standard[2]. Three types of PIN are distinguished: assigned derived PINs, assigned random PINs and customer-selected PINs. To the customer the only detectable difference is between a PIN he is assigned or a PIN he can choose himself.

Assigned PINs are usually numeric and typically have between four and six digits. In the UK, for example, PINs of four decimal digits are widely used. In practice, PINs are not usually memorized, as they should be, but written down, sometimes on the card itself. It is not so unsafe to write the PIN in an obscure way in some part of a diary but, if the bank were to suggest this and give examples, a resourceful crook would know which schemes were widely used. Ultimately, the bank has to depend on the good sense of its customers and recognize that the security which derives from separating the PIN from its card is at best only partial.

Since many people now find that they have been assigned many PINs for different purposes, memorizing these PINs becomes difficult. One of the advantages of customer-selected PINs is that the customer can choose one value and use it for all purposes. There seems to be an advantage in alphabetic PINs to make a memorable word and thus avoid the need for writing it down. Alphabetic PINs do not imply an alphabetic keyboard because a ten-key layout of the form shown in Figure 10.7 is used — this is the proposed American National Standards Institute (ANSI) standard and is based on the telephone keypad. Unfortunately there is support elsewhere for a layout based on the calculator keyboard, where the top row of keys is 7, 8 and 9.

There is clearly a need for careful guidance to customers. Where the customer selects his PIN he should be given time to think about it, not required to choose it when the account is opened. This can be done by assigning a temporary PIN which makes the ATMs usable at once and then allowing him to change this to his own chosen value when he wishes. The card issuer can accept the customer's chosen value by post or telephone if he is given a serial number unrelated to his account (or temporary PIN) with which he can validate his new choice. It can

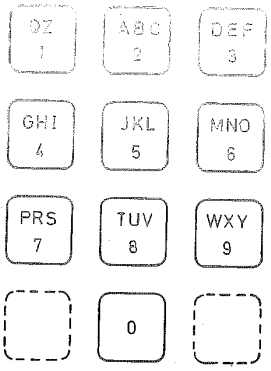

Fig. 10.7 Alphanumerical PIN pad layout

easily be arranged that no employee need see the association between the customer's chosen PIN and the account number or name and address.

Any system employing such a short password as a four decimal digit PIN must be careful to prevent methods of discovering the PIN by searching through all of the 10 000 possibilities. On-line systems make this restriction easy. Typically, ATMs allow three attempts to give the correct PIN, then refuse payment. Early systems retained the card after the third incorrect try but if an ATM card also functions as the credit card this card retention is very unpopular. In an off-line system it would seem dangerous to give the card back because it allows PIN searching to continue.

An excellent source of advice on PIN management is the *PIN Manual* published by MasterCard International[3] which is reproduced as chapter 10 in *Cryptography* by Meyer and Matyas.[4]

Algorithmic PIN checking

When off-line operation is used or when an on-line system must be capable of checking PINs in a temporary off-line state, the method of checking must depend on an algorithm, typically a cipher with a secret key. Such methods of PIN checking are described as examples in the ANSI Standard. Figure 10.8 shows the method by which a (small) PIN of N decimal digits can be derived from the customer's account number. First the account number is padded to 16 decimal characters, using zeros or some other constant value, then the resulting 64-bit number is enciphered using a secret DES key. From the 64-bit result a decimal number of N digits is derived.

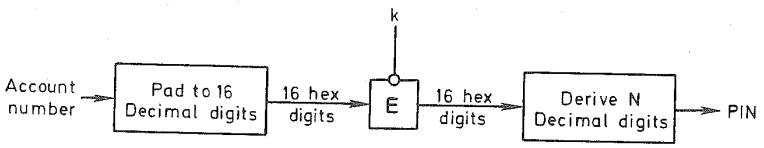

Fig. 10.8 Derivation of PIN from account number

The method of deriving the decimal digits is to examine the 64-bit result 4 bits at a time, starting from the least significant end and accept only those 4-bit groups whose binary number is less than 10. The accepted binary numbers form the decimal digits. This is similar to the method by which decimal digits are derived for cipher feedback (see page 92).

The PIN management standard anticipates PINs of between 4 and 12 decimal digits, but typical PINs have 4, 5 or 6 decimals. In most cases these will be derived from the 16 hexadecimal digits comprising the 64-bit output of the DES. If, by bad luck, insufficient digits are generated, the remaining hexadecimal characters can be used, subtracting 10 from their binary values to get a digit value lying in the range 0 to 5. The very small bias introduced in this way is generally acceptable.

This method generates an assigned PIN. If a customer-chosen decimal PIN is needed, the same procedure is used to derive an N decimal number, then the chosen PIN is added to the derived PIN, by decimal arithmetic without carries (individual digits are added modulo 10). The resulting decimal number is called an 'offset' and this is stored on the card. Because of the approximate randomness of the derived PIN, it is not possible to deduce the chosen PIN from the value of the offset.

When a card encoded in this way is used with an on-line ATM, PIN checking can be done in one of two ways. The offset can be read from the card with the account number and the PIN checked algorithmically at the terminal, or the PIN can be transmitted to the centre and checked against a file of chosen PINs.

Instead of storing an offset on the card an alternative is to store a number derived cryptographically from the PIN, which is called a *PIN verification value* or PVV. Because the range of PIN values is so small, an attacker could make a table showing the relationship of the PIN and PVV, so in practice the PVV is made a cryptographic function of both the PIN and the account number. This is illustrated in Figure 10.9 where the derivation is by encipherment, followed by producing N decimal digits in the way described earlier. In practice the encipherment uses a double length key because this key has a long life and might be subject to prolonged attack. The PVV can be recorded on the card and used for local PIN checking, but since a compromise of the key would be disastrous, this kind of checking should only be performed in a very secure environment. Alternatively, if the PIN is sent, under cryptographic protection, to a central place, it can be checked against a file of PVV values, which is safer than holding the PINs themselves.

Typically the PVV has four decimal digits, but this has a disadvantage. Suppose that an attacker is making a search of PIN values to find the correct one. With a four-digit PIN he will obtain the correct value with probability 1/10 000. However, the PVV is a random function of the PIN hence there may be other

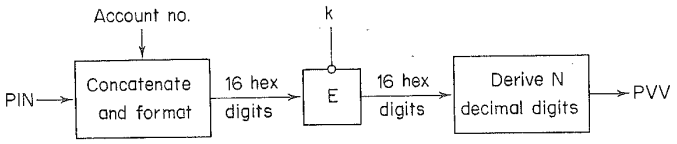

Fig. 10.9 Derivation of a PIN verification value

PIN values which give the correct PVV. The average number of PINs associated with a valid PVV value is $e/(e-1)$ which is approximately 1.58. This increased probability of finding an acceptable PIN (if not the correct one) makes the PIN searching attack somewhat easier. This effect can be reduced to a negligible one by employing a PVV field of five digits, when the number of PINs corresponding to a valid PVV is on average 1.05.

The dialogue for an on-line ATM

The essential transaction between the Automatic Teller Machine (ATM) and its central data base contains two messages, a request from the ATM to the centre for authorization of the transaction and the response from the centre which is either an authorization or a denial.

Typical contents of the request message are

ATM identity,
Account number and other identifying information from the card, such as a card serial number,
Watermark or other security number read from the card,
PIN provided by customer, enciphered,
Amount of cash requested,
Sequence number of transaction,
Authenticator for all the other data.

The response could contain the following

ATM identity,
An operation code signifying payment approval or denial, or denial with card retention,
Sequence number of transaction,
Authenticator for all the other data.

Both messages need authentication otherwise the request could be modified to reduce the amount debited to the account or the response could be altered to change a denial into an authorizaton. Authentication could consist of enciphering the entire message since the serial number, if managed correctly, prevents the substitution of a previous message. The essential cryptographic requirement is to encipher the PIN so that a line tapper cannot discover which PIN value is associated with a given account.

In spite of all precautions the response message might not get through, then the customer would suffer because his account would be debited but the money would not be paid at the ATM. To help resolve these problems, a log is kept at the ATM, sometimes in the form of a printed tally-roll. Such a log is also needed to satisfy the internal accounting of the bank when the ATM's payments are reconciled with the cash loaded into it.

It might be thought that greater certainty could be obtained by a third message, from the ATM to the centre, confirming that the payment had been made. The lack of this message would signal to the bank a possible discrepancy, without waiting for the customer's complaint. This would enable the rare examples of system failure to be detected earlier. But no sequence of messages in the dialogue

can provide certainty, because the final message could be lost and there would still be doubt about the outcome. This is an example of the impossibility of final agreement when messages are exchanged by a channel which is not perfect. This is known as the 'Two Generals Problem' and has been studied carefully, but there is no effective solution. Whether a two-message or three-message dialogue is used affects the state of knowledge at the centre but does not solve the basic problem. A permanent record at the ATM is essential and this should if possible resist forgery attempts, in order to reduce the possibility of employee fraud. If a magnetic recording is used, authentication or digital signature on the record can provide this additional security.

This dialogue of two or three messages is only one possibility. There is another scheme in which the PIN is not sent from the ATM in the request but, instead, an 'authentication parameter' is returned with the response. This is described later on page 309 in the context of shared ATM systems.

The essential requirements of message authentication and PIN encipherment can be provided by any of the standard methods described in earlier chapters. A less conventional method for PIN encipherment and approval authentication is described as an example in the PIN Management Standard. This is illustrated in Figure 10.10. It employs a decimal counter in the ATM which is reset only when the keys are changed. The output of this decimal counter is enciphered twice to generate 32 hexadecimal digits from which are derived, by the method described earlier, sufficient decimals to encipher the PIN and a further 5 decimal digits which become the approval authentication code. The PIN is enciphered by modulo 10 addition of the generated digits.

At the other end of the communication line a similar algorithm is employed with its counter in synchronism and, to keep the counters synchronized, two or three digits of the counter are transmitted with each request message. At this central place, the same $N+5$ decimal digits are derived, where N is the number

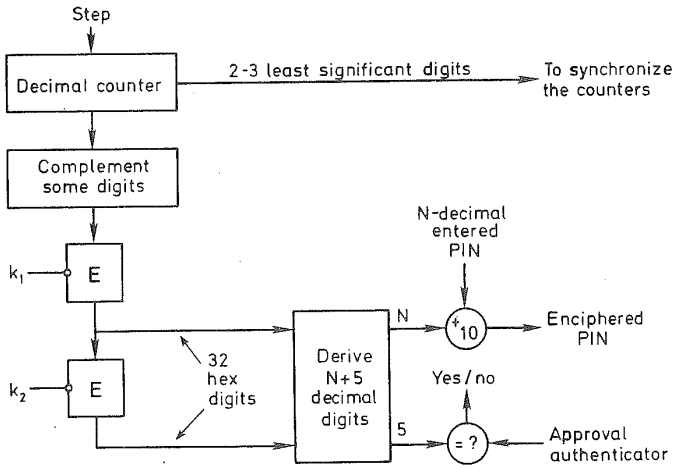

Fig. 10.10 Method for PIN encipherment and approval authentication

of digits in the PIN. By subtracting these values, the PIN is deciphered and checked against the local records. If the payment is authorized, the 5-digit approval authentication code is returned in the response message. The method does not handle the authentication of the request message, which is an essential security requirement.

We should consider the ATM and the central system as 'communicating entities' and consider how they authenticate each other in one of the ways described in chapter 5 (see page 126). The sequence number from the ATM is returned in the response by the centre and both messages have authenticators; this authenticates the centre to the ATM and prevents intrusion by a recorded earlier message. The sequence number must not repeat during the life of the authentication key. The centre authenticates the terminal by holding the sequence number last received from each ATM and requiring the next one to be incremented correctly.

The security of this sytem depends on the secrecy of keys — a different key for each terminal — the integrity of the ATM software and hardware and the integrity of the software and database at the centre. Since the ATM is physically secure, the biggest dangers at the terminal come from those who are allowed physical access. Preferably, the electronics should not be accessible to bank employees and critical parts of the electronics should be repaired only by replacement under strict control. At the centre, PIN files should be secured by encryption. The control of card and PIN issuing, and the recovery of lost PINs, must be carefully designed so that no single employee (or small group of employees working closely together) has access to enough information to reconstruct corresponding PINs and card data.

Shared ATM systems

When a group of banks join together to make all their ATMs available to the customers of the whole group, the task of approving payments takes on a new dimension. It is unlikely that the methods the various banks use for verifying PINs against card identity will be the same, therefore no single algorithm operating in an ATM can verify the PINs of all the customers it serves. Banks could not be expected to trust other banks with their secret keys for verifying PINs and it would not be practicable for lists of all the PINs against card identity to be stored in all the banks. From this it follows that PIN verification must be done centrally by each card issuer, that all the ATMs must be on-line when they serve customers of other banks, and that the ATM processors of all the banks must be connected by a communication network.

Within this network the banks must exchange authenticated messages, they must be able to carry PINs securely from one bank's processor to another and they must maintain accounts of the transactions carried out which are satisfactory to all the banks, so that settlement can take place for payments that have been made by one bank to another's customer. All this implies the exchange of secret keys between the banks for authentication.

Many schemes for PIN encipherment and message authentication in shared networks have been proposed. In the context of point-of-sale systems (treated later in this chapter) methods using session keys have been described[5,18] and also methods using personal keys stored on the customer's card.[6] Some shared ATM

networks use a system which makes extensive use of system keys, known as 'zone keys' and is similar in concept to 'node-by-node' encipherment, described in chapter 4 (see page 106). All methods have some advantages and disadvantages. The method we shall describe' is called 'zone-encryption' and it is resistant to customer fraud and line tapping. It requires very careful handling of the system keys so that bank employees cannot discover keys of their own or other banks. Later, we shall show how the intelligent token (or smart card) and public key encipherment and signature can, separately or together, improve the security of shared systems against employee fraud.

Typically, in the zone-encryption method, a group of banks appoint an authority to operate on their behalf for the interchange of messages for ATM payment approvals. This 'interchange' is represented by a processor which acts as a node to which all the processors of the banks involved in the ATM system are connected. In general a bank is both an issuer of cards and an operator of ATMs. When a customer goes to a bank which is not his own, that bank becomes the *payer* of money to the customer. The other bank involved in the transaction is the *card issuer*. The issuer holds the account that is to be debited as a result of the transaction and must credit the payer bank during the settlement process. Figure 10.11 shows the links involved. The ATM belongs to bank A which is the payer and the ATM is connected to the processor HA which is in turn connected through the interchange I_1 to the processor HB of the bank B which is the card issuer. The request message from the ATM takes the path ATM-HA-I-HB to the card issuer. The PIN is verified at HB against the card identity, the status of the card and the account are checked and a response is returned giving or denying approval for the payment. The approval response has two functions, it allows the ATM to issue the cash and it also allows the payer bank A to present an item in its account for settlement by bank B. In the figure we also show a second interchange I_2 because we assume that shared ATM systems will grow by the joining together of interchanges to generate, eventually, very large systems. The principles which are adopted for shared ATM systems should be capable of extension to a hierarchy of sharing.

The system must keep to a minimum the opportunities for fraud by employees of one bank against another. Clearly the banks must share secret keys if symmetric ciphers are used for authentication and encipherment. These keys must be handled in a way which reduces the dependence of one bank on another's

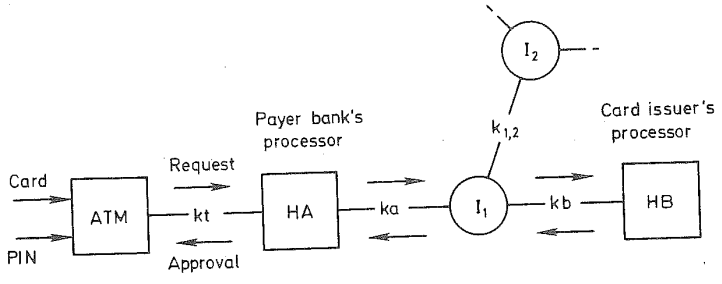

Fig. 10.11 Messages in a shared ATM network

security. In the figure, each link is associated with one secret key. The various ATMs attached to processor HA can have individual keys of which *kt* is an example. For connection to the interchange, bank A uses key *ka* and bank B uses key *kb*. Between the interchanges I_1 and I_2 the key $k_{1,2}$ is used.

The PIN entered by the customer at the ATM is carried in the request message to the processor HB and the approval is returned via the payer bank's processor HA to the ATM. The encipherment of the PIN in the request message, the authentication of the request and authentication of the response employ the keys shown on each link, therefore a transformation of the request message to a new key takes place at processor HA and at the interchange I_1.

A transformation at the interchange is necessary if a large number of banks is involved and it is not convenient to establish a key for each bank pair. Dividing the network into 'zones' in this way is essential if it is to be capable of unrestricted growth by combination with other banks or groups. The transformation at HA is not strictly essential, because bank A might decide to use a single zone-key *ka* for all its ATMs and for interchange. It could be argued that this increases the risk because a discovery of one key then exposes the entire network of bank A. In principle, discovery of *ka* exposes the network even when each ATM has it own key, *kt*. Probably the storage of *ka* is better protected than that of the keys *kt* in the terminals, but this is arguable, since ATMs are physically strong. For generality we shall assume separate terminal keys.

Whenever a message is transformed between one zone-key and another, as in the interchange, there are potential security risks. Keys of two different banks are present and might be exposed, the PIN is deciphered before re-enciphering and thus appears in plaintext form and the authentication of the message is renewed — a process which is vulnerable to active attack. Such a procedure cannot be safe in a mainframe processor and it is, therefore, recommended strongly that all the sensitive data and functions are within a physically secured module. Figure 10.12 shows what happens to a message inside such a module.

Keys for all the links entering the node are held inside the secure module, but if there are too many keys they can be called in from storage outside, on condition that the keys are enciphered by a 'storage master key' while they are outside the module. In the example of Figure 10.12, the key k_1 is associated with the incoming message M_1 and the key k_2 is associated with outgoing message M_2. First the authenticator A_1 of M_1 is checked using k_1. If this fails, the message sent forward is not the request or response message but a diagnostic message to state the nature of the error. It is sometimes suggested that a failed message be sent forward with a wrong value of authenticator, to propagate the failure, but this loses useful fault-tracing information. If the authentication process checks, a new message is assembled using the PIN value which has been deciphered with k_1 and re-enciphered with k_2. There seems no good reason to use different keys for the encipherment and authenticator procedures. The new message is equipped with a new authenticator A_2, assembled and sent forward.

Instead of separate encipherment and authentication, the whole message can be enciphered. If precautions are taken against reply or resequencing of messages this is equivalent to authentication. It has been argued that encipherment makes treatment of the request message at the card issuer more difficult because the whole message must then be deciphered for authentication, and the PIN is

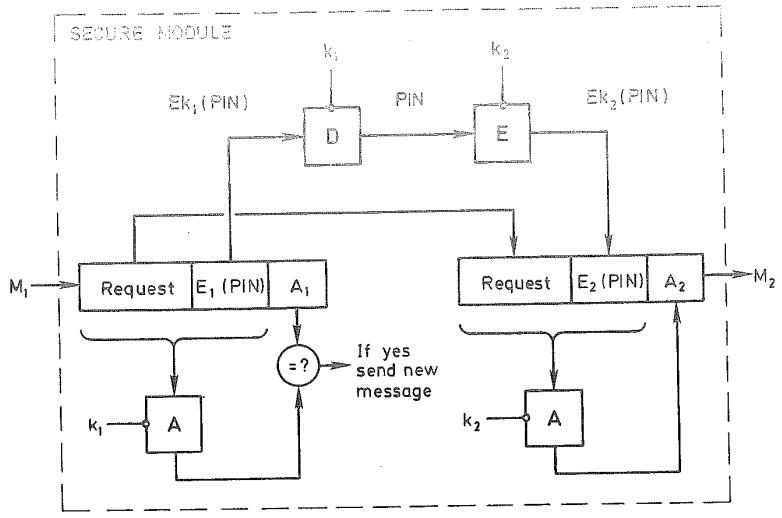

Fig. 10.12 Message conversion between zone-keys

revealed. But if the decipherment takes place in a secure module where PIN checking is done this does not reduce security.

The precautions against replay or resequencing require that message sequence can be checked at the receiver. The ATM's request messages can easily maintain their own sequence counters so that all messages they authenticate are different, but the checking process at the issuer will not be able to keep a note of the sequence numbers of all sources if there are very many banks in the interchange. In this case a date and time field is the best solution, with the precautions already noted in chapter 5. Authentication between a bank and its own ATMs should be able to use sequence numbers.

There is a different way to combine authentication and encipherment using the same key which is to compute an authenticator in the usual way over the message (excluding the PIN) and add this to the PIN before incorporating it in the message. The addition could be modulo 2 addition of binary digits, or modulo 10 addition of decimal digits, according to the way the authenticator is coded. At the receiver, the authenticator is recalculated and, by subtraction from the authenticator in the message, the PIN is obtained. If PIN checking fails the cause may be either a wrong PIN value at the ATM or an authentication failure. Perhaps it does not matter, since the outcome is to refuse payment in either case. Figure 10.13 illustrates this method and Figure 10.14 shows how easy it is to recompute the authentictor/PIN at the interchange or wherever a new key is needed.

Each bank must trust the other banks in the interchange system to the extent that their handling of keys is safe and the physical protection of the secure module is good. Each bank has within its grasp sufficient information to derive the PINs of transactions it handles. But this is inherent in the use of PINs, since they are in any case exposed at the point of entry into an ATM.

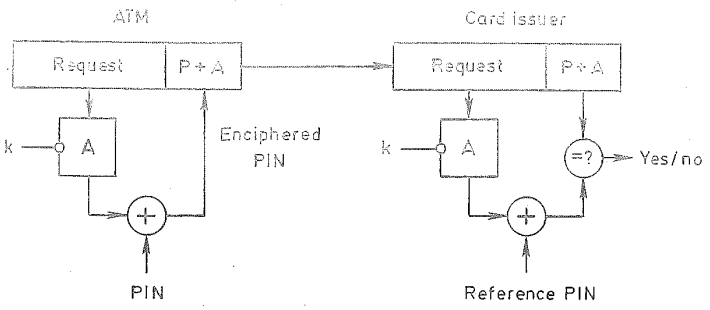

Fig. 10.13 Combination of authentication and PIN encipherment

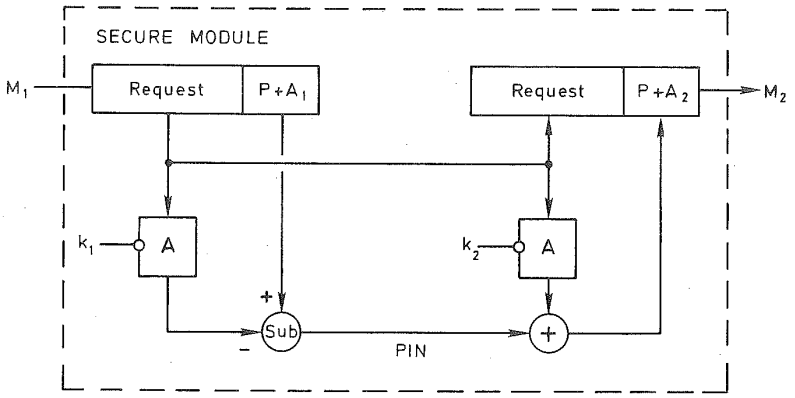

Fig. 10.14 Message conversion for P + A method

Checking the PIN with an authentication parameter

An alternative method of checking the PIN is to transmit from the card issuer to the ATM a function of the expected PIN sufficient to make the check. This quantity is called an 'authentication parameter' (AP). Suppose the pin value is P and there are data D on the card's magnetic stripe, including the account number and other things. Some of these extra data may deliberately be made 'random' or unpredictable. The authentication parameter A is a complex function $A = f(P, D)$. For example, the function might be derived from data X related to the account number and other card related data Y. A suitable function f would be $X +_2 Ek(X)$ where $k = Y +_2 P$, since this is a one-way function of both X and $Y +_2 P$. The reference value of A is held at the card issuer's processor. When it receives a request containing the account details for this card it first verifies the account status. If this is satisfactory it authorizes the payment in its response, subject to the correct PIN being given and returns the response value A for the check to be made. At the ATM, when the PIN is received, with X and Y derived from the card, a local value A' is calculated. If $A' = A$ the payment is made and a

confirmation message is returned to the card issuer, which records the transaction and updates its accounts. This third message is essential if the AP method is used, since the outcome is still uncertain when the card issuer gives its response.

The message in which the AP goes to the ATM can state other PIN criteria, such as the number of tries the customer has, to get the correct PIN, and whether the card is to be retained if all fail. The trials can be conducted locally, whereas the other method, in which the PIN goes to the centre, requires network messages for each trial. The number of trials must be a parameter in shared ATM systems because there are several issuers and each sets his own rules. Some may not even use a PIN, and in that case the card issuer does not want the ATM to ask for one. So the AP method is more economical in terms of network messages if there are in practice enough re-tries of the PIN.

The AP method of PIN checking does not remove the need to transmit data in secret. Knowledge of A does not allow $Y+_2 P$ to be determined directly, even if X can be deduced from the account number in the message. But if Y is also known it is usually easy to find P because of the restricted range of passwords in bank systems. Many use only 4 decimal digits, giving a task of only 10 000 trials to find P. The card formats used by different banks vary and some have little data Y that are not deducible from the account number. A general purpose network able to handle messages for many banks should regard the secrecy of the AP just as seriously as that of the PIN. The obtaining of the PIN value by a search is not a result of the particular form of AP function employed. It is due to the small range of PIN values, since any 'one-way function' can be inverted if its range is small.

Even allowing that there is little security advantage, the AP method can have advantages in limiting the worst-case number of message per transaction. The 'confirmation message' which goes from ATM to card issuer is an overhead which some consider necessary to help resolve the cases where messages are lost. Some packet-switched networks allow three messages for a unit fee in the 'fast select' mode, so the confirmation message may be 'free'.

Public key cryptography in a shared ATM system

The security requirements of enciphering PINs and authentication messages can be met by public key ciphers and signatures.

Taking first the case of a single ATM network, it is only necessary to have one set of keys, a secret key at the centre and a public key which is stored in each ATM. This allows encipherment of the request and confirmation messages and signature of the response from the centre. Secrecy of the response is not needed unless the AP method of PIN checking is used. A further requirement is to authenticate the request message, since otherwise an active attack could modify the request or introduce bogus requests. To avoid the need for the ATMs to hold their own secret signature keys, they can use a password to identify themselves, since this is concealed in the request message. A random number in the request, which is returned in the response, prevents the ATM acting on a replayed response. If a second random number is used by the centre to authenticate a confirmation of payment, an active attack can be detected even if the password were discovered.

In the context of shared ATM systems, public key methods are even more valuable because they avoid the need for zone keys and the corresponding key distribution problems. But the simplifying feature of using the ATM's password for authentication is lost. The ATM must be able to sign its messages using its own secret key. A group of banks in a shared system can easily exchange their own public keys but it would be difficult, even impracticable, for them to exchange the public keys of all their ATMs. This problem can be solved by having each ATM send with its request a 'certificate' of its public key signed by its own bank.

If the shared system becomes very large and expands rapidly by combining with other systems, keeping in each ATM the public key of each issuer becomes difficult. Then a central authority can be established as the key registry and each card contains the public key of its issuer, signed by the registry. The ATMs hold the registry's public key and check the issuer's public key before using it to encipher the (signed) request message.

There is a more radical approach which embraces shared ATMs, point-of-sale EFT and other payment systems, employing a 'smart card' to originate a transaction with a digital signature. This is described in section 10.6 starting on page 329.

10.5. POINT-OF-SALE PAYMENTS

The majority of payments in shops may be cash but it is the cheque and credit-card payments that concern banks. These present two problems, the expense of operating them and the increasing extent of fraud. The operating cost can be reduced by improved data capture, avoiding paper documents. The fraud can be reduced by on-line operation and improving the identification of the cardholder. The PIN, though it has limitations, has proved reliable enough for ATMs so its extension to point-of-sale payments seems inevitable.

From the customer's viewpoint he has not much to gain. Having a different PIN for each of several payment cards will be a nuisance. Moving from a credit scheme to one in which his account is debited at once is a disadvantage. Neither of these things is inherent in point-of-sale EFT but they are possible consequences of the change. The shopkeeper can gain from the more rapid payment and because the discount to the credit-card company is no longer needed. Hopefully the salesperson will find the procedure easier and the responsibility for checking the card signature and validity is ended. The operation should be faster, which helps at busy situations such as supermarket checkouts. The balance of advantage is in favour of extending the use of EFT in shops and phasing out the paper-based systems.

This is the motivation for point-of-sale EFT. The basic principle is illustrated in Figure 10.15. The main parties in the transaction are the customer who holds a card (or other token) and remembers his PIN and the merchant who is to receive the payment for goods or services he supplies. In general these parties have accounts at different banks, the customer relating with the 'card issuer' and the merchant with the 'acquirer'. The acquirer will usually have installed the terminal and connected it to its own payment system. In the figure, the terminal is shown connected to the acquirer's system HA and the acquirer's system exchanges messages with the card issuer's system HB through an interchange. The figure shows only one of several message patterns which are discussed below.

Typical operators of point-of-sale terminals will be those who need to receive payments of modest amounts from the public, such as shops, travel agents, ticket offices and professional services. No one word is adequate to denote this wide range of activity so we will employ the conventional term 'merchant' to designate the person or institution which uses a point-of-sale terminal to receive payments. Whereas an ATM system for a single bank can flourish in the countries where there are nationwide banks, point-of-sale systems are not attractive unless the merchant can use his terminal to accept cards from a large group of card issuers. They are essentially shared systems.

The sequence of operation is that a sale is agreed between the customer and merchant and the payment amount is displayed for the customer's approval. The customer then passes his card through the card reader on the point-of-sale terminal (usually a simple swipe reader) and enters his PIN on a keyboard attached to the terminal.

Before the transaction can be completed a payment-request message must go to the card issuer who will verify that the card is valid, the PIN correct and the account in good order. A response then travels from the card issuer HB to the acquirer HA. This 'approval' response is effectively a payment message which authorizes the acquirer bank to credit the merchant's account with the amount requested. From the acquirer's system an advice passes to the point-of-sale terminal telling the merchant that the transaction is complete and causing it to print a receipt for the customer.

The natural progression of messages is from terminal to card issuer to acquirer to terminal — a triangular route. The first requirement is to validate the card and customer, the second to make a payment from HB to HA and the third for HA to tell the merchant that the payment is made. But a triangular path makes it very difficult to recover if the transaction fails in the middle. Recovery must at any one time be the responsibility of one unit in the system. We can assume this is not the terminal because of its relative insecurity. The scheme of Figure 10.15 places the acquirer's processor HA in the strategic position to handle recovery by re-tries. For this purpose, the outgoing message to the card issuer goes through HA, where the new transaction can be logged for checking its safe return as a payment or a refusal. This pattern of messages also matches the bidirectional nature of most data calls in a switched network.

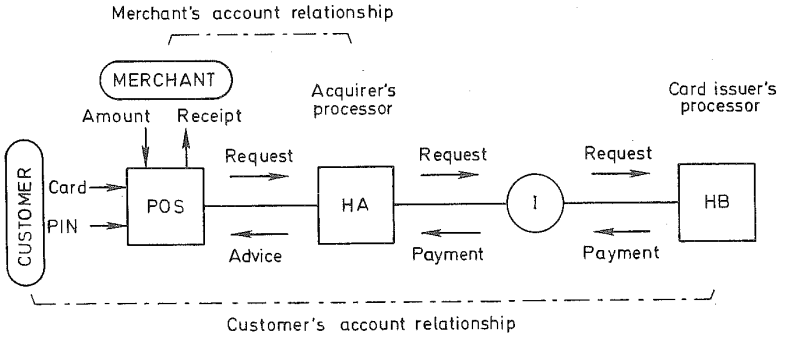

Fig. 10.15 Point-of-sale electronic funds transfer

This scheme requires both the card issuer and the acquirer to be on-line for a transaction to happen. Central checking of the account certainly requires the card issuer, but the actual payment can wait, provided that the merchant can quote the message from HB stating that a payment has been recorded. Where the number of issuers is small and if there is good authentication in the messages, the card issuer can take the central role and the acquirer's participation can be delayed. HA and HB change position in Figure 10.15. Then the onus of recovery falls on the card issuer's processor HB. The message path is terminal — card issuer — acquirer and back to the terminal. In yet another adaption of the system, the responsibility for recovery can fall on a separate processor working for the whole network, much as I does in the figure. There are many ways to organize the responsibilities and the message paths, but the basic security requirements are similar in each case.

In addition to these normal payments there are other types of transaction which the terminal must generate. Since the use of the terminal must be controlled, it is logged on and off by the merchant using a magnetic-striped card and PIN for the purpose. This transaction is verified at the acquirer's system and it may, in some systems, result in the distribution of session keys to the terminal. Another, optional, transaction type is the *refund* which requires special authorization, by the same merchant's card and PIN.

The payment messages can cause instant debiting of the customer's account and crediting of the merchant's account. These immediate transactions are balanced by equal amounts in accounts maintained between the issuer and acquirer. For example, the issuer may maintain an account for the acquirer which he credits with the amount paid. These accounts are settled periodically by an inter-bank transfer.

Point-of-sale EFT provides a better method of capturing data and helps to reduce fraud, but it changes the nature of the transaction by debiting the customer's account at once. Many customers will find this unwelcome and prefer to continue with cheques and credit-card transactions where the delay in payment works in their favour. It is not essential that these systems debit accounts at once. They capture the data and check the card validity in a way which could equally function like a cheque, with a clearing delay, or a credit transaction.

The requirements for security in point-of-sale systems are essentially those we have already examined for shared ATM networks. For on-line systems they can be summarized in two statements.

1. The verification of the PIN entered by the customer must be carried out in association with the issuer's processing system. The PIN value or AP value carried through the network for this purpose must be enciphered to protect it against line tapping or disclosure in any of the processing systems through which it passes.
2. The messages which carry the request and the approval/payment response must be authenticated to prevent their undetected alteration whether by active line tap or in any of the processors through which they pass. Among other attacks, the possibility of a replay must be guarded against.

The principles we have described earlier show us that both the encipherment and the authentication must employ variable elements and the receiver of

messages must be able to check the sequence in which they were sent. Any of the authentication methods described in chapter 5 will serve.

If the 'zone key' principle is used there is a different key for each of the three links shown in the figure. The presence in one processor (for example the interchange) of keys shared with a number of other institutions presents risks which can be minimized by using tamper-resistant modules (as we described for the shared ATM network), and by paying careful attention to key distribution. Employees of the interchange organization should have no means of access to the keys installed for communication with their associated banks.

Figure 10.15 shows the messages carried for a normal payment in the chosen message scheme. The request message travels from the terminal to the issuer's processor HB. The intervention of HA and the interchange I in this message transport is needed for the change of keys when the zone encryption scheme is employed. The response message travels from HB to HA as a payment and continues on to the terminal as an advice to the merchant that the payment has been made.

A refund transaction needs special authorization at the terminal, employing a card and PIN belonging to the merchant. In this case the request message carries the PIN, enciphered, to HA where the PIN is verified and the message authenticated. If these are correct, the payment message is sent from HA to HB. The advice then returns from HB via HA to the terminal. The requirements for PIN encipherment and message authentication are similar to those for normal payments.

Point-of-sale payment terminals are part of an interchange system shared between many card issuers and acquirers so that a wide choice of cards may be used. Since it would not be practicable for the card issuers to exchange their blacklists of lost, stolen and otherwise exceptional cards, there is a strong argument for on-line connection of terminals. Nevertheless, the cost of on-line connection and the communication difficulties in some countries will strongly favour off-line operation. Off-line terminals collect together the records of payments and periodically the complete record of payments is sent to the acquirer bank, either as physical record or by a telephone call.

A compromise between the cost of on-line connection and the need for security is to use off-line transactions for payments below a certain value and within a participating group of banks. For payments of high value and those involving other banks, a validation call is made to a service operating on behalf of the card issuer.

The use of public key signatures promises to improve security and increase convenience in payment systems generally. A comprehensive payment system using public keys can be devised which is suitable for large ATM networks, large point-of-sale networks and off-line transactions between bank customers analogous to today's bank cheques. In the final section of this chapter we shall introduce this form of payment using signed messages and show how it can be adapted to a point-of-sale network.

The transaction key method

In Figure 10.15, the handling of keys for point-of-sale EFT is like the method customarily used for ATMs but there is a world of difference between the physical

security of an ATM and of a point-of-sale terminal. Because of the need to protect the money it holds, an ATM is made strong and can physically prevent anyone from reading the secret keys it holds. On the other hand, when point-of-sale terminals are widely distributed in shops a criminal gang can easily obtain a few, take them apart and discover, after some effort, how to read the keys contained inside. They might, in the course of reading the key, virtually destroy the terminal so that it cannot be used again. Even so, if they had recorded the transactions for a long period before breaking into the terminal, possession of the keys could give them all the PIN values that have been exchanged. Terminals can be made more secure by further tamper-resistance features but, in the end, their keys can be read out.

To make point-of-sale EFT systems more secure, special forms of key management have been developed, specifically to prevent the kind of attack we have described, called *back tracking*. There is another kind of attack called *forward tracking* in which the terminal is put back into use after the keys have been discovered and the subsequent messages it handles betray the PIN values. But anyone who is skilled enough to do this could probably arrange for the terminal to record for him all the PIN values as they are entered. Therefore, forward tracking is such an advanced attack that, even if only back tracking is prevented, the system is safer.

A scheme of key management which, if the circumstances are right, prevents both forward and backward tracking is *transaction key*. The principle was first explained by Beker, Friend and Halliden[7] in a form in which each terminal maintains a special key in common with each card issuer and this key, the transaction key, is changed at every transaction in a way in which an enemy would find it hard to follow. This is an excellent principle, but when there are very large numbers of terminals and a fairly large number of card issuers, the multiplicity of keys makes it difficult to administer. In particular, as terminals come on line or are replaced after maintenance, all the card issuers have to be updated with their transaction key data. For this reason, most practical applications of the transaction key principle employ the transaction key principle between the terminal and an *acquirer*, with each acquirer having to handle only its own set of terminals. The scheme has been extended to protect the PIN while it continues in transit to the card issuer, as we shall describe below.

There are two special features of the transaction key method, the way in which it handles the authentication and integrity checking of messages and the way in which it maintains the transaction key and modifies it after each transaction.

Authentication and integrity checking are provided by the MAC based on the data encryption standard. When a MAC is produced the results of the cryptography are 64 bits from the final DES operation. Normally, only the left-hand 32 bits are used as the MAC and the right-hand 32 bits are discarded. In the transaction key method the right-hand 32 bits (called the *MAC residue*) form an important part of the scheme. We shall abbreviate MAC residue as MAR. When the terminal sends a request to the acquirer, it includes the MAC in the request but retains the MAC residue (MAR1) for use in authenticating the response. When the response is returned, a MAC is included with it and the MAC residue (MAR2) is retained by the acquirer. Each checks the MAC value on the messages it receives. The special feature of the transaction key method is that the MAC attached to the response is formed cryptographically not just from the message but from the

message preceded by MAR1. The acquirer has obtained MAR1 as a byproduct of checking the authenticator on the request. But MAR1 is not sent in the message, it is, so to speak, a 'ghostly' part of the message and the terminal will not be able to check the authenticator unless it precedes it by MAR1 before applying the cryptographic process of MAC calculation. As we shall see later, MAR1 is not the only ghostly part of the response message, there may also be an authentication parameter (AP) which we describe later. This idea of including in each message a ghostly residue of the previous message before authenticating it effectively chains the messages together and assures the terminal that the acquirer has properly authenticated the request. If more than two messages are used in the dialogue, for example there may be a confirmation from terminal to acquirer, the same chaining principle can be used, the response having its MAC formed with a ghostly value of MAR2 preceding it.

Perhaps more important than this authentication scheme is the way that the transaction key is formed and used. For this purpose the terminal maintains a 56-bit *key register* in the DES format for each acquirer it must work with and the acquirer holds the same values, having one stored for each terminal with which it communicates. After initialization to the same values, the key registers (KR) must synchronize their values after each transaction so that the terminal and acquirer can continue to communicate.

Figure 10.16 shows how the key register is used at the terminal and how it is updated. The function shown as O is a one-way function $O(k, x)$ derived from the DES operation as

$$O(k, x) = Ek(x) +_2 x$$

The symbol we use for 'O' shows which is the key input by the small circle.

From KR and data read from the card tracks, two transaction keys are formed tk_A and tk_E. The first of these is used for authentication and the second for enciphering the PIN between the terminal and acquirer. The input DC1, DC2 and DC3 are all derived from card data. DC1 and DC2 come from the account number and are therefore known to the acquirer but DC3 uses some of the discretionary

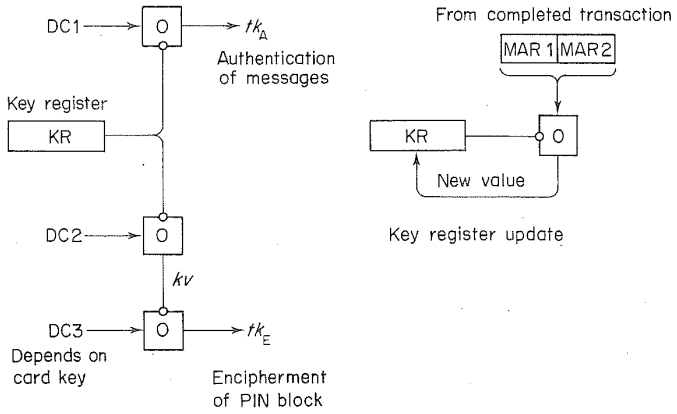

Fig. 10.16 Transaction key production and key register update

data field together with the account number for its calculation. Data from the discretionary field on the card is not sent with the transaction and is not known to the acquirer. The purpose of forming tk_E in this way is to encipher the PIN using a key which can normally only be calculated by the card issuer from its knowledge of the contents of the discretionary data field. Note that the quantity *kv* is known to the acquirer. This preparation of tk_A and tk_E occurs at the beginning of the transaction and these values are used throughout. At the end of the transaction the KR value is updated in the way shown in Figure 10.16, using as input a concatenation of the two MAC residues which are known to both terminal and acquirer at the end of the transaction. When there are more than two messages in the dialogue, there are special rules determining which MAC residues shall be used. After this update, KR depends on the entire progress and outcome of the transaction. If an attacker knew how it was initialized and wanted to track its value he would have to record every message and also discover the discretionary data from which DC3 was derived. The value DC3 is sometimes known as the 'card key' because it has a degree of confidentiality to the terminal and card issuer. It is not difficult to read it from the card but the need to maintain an unbroken chain of complete data including the card key makes forward tracking very difficult. Backward tracking is impossible because of the one-way function.

We have seen how tk_A is used to authenticate the request and response. The enciphered PIN must reach the card issuer who does not have the means for maintaining a KR value. To enable the card issuer to reconstruct tk_E in order to decipher the PIN, the acquirer calculates *kv* and sends it to him in the request message as shown in Figure 10.17.

There is an implication in the way that the scheme is officially described that *kv* should be protected by encipherment for transit between acquirer and card issuer, though this is not mandatory. Certainly, if *kv* is known and the card key, the PIN is not safe from decipherment. Depending on the level of security desired users may wish to encipher *kv* and this encipherment uses a conventional key which is changed from time to time but not per transaction.

In order to authenticate that the response came from the card issuer, an authentication parameter (AP) is returned from the card issuer to the acquirer when it

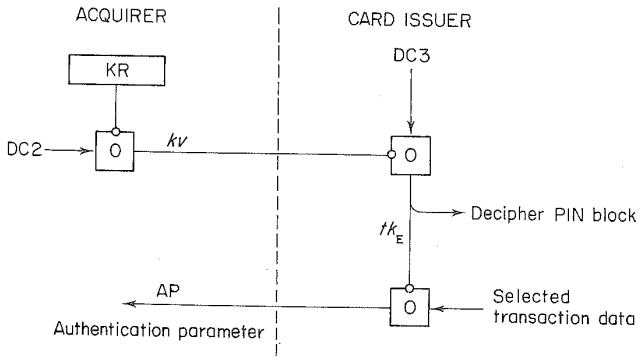

Fig. 10.17 Authentication of card issuer response

approves (or not) the transaction. This is formed by operating with k_2 using the one way function on certain critical data taken from the request message. If the AP is correctly returned, the transaction is assumed to be approved. In order to confirm this from acquirer to terminal, AP is concatenated with the response message when its MAC and MAR are formed, though it is not returned in the response message sent. Together with MAR1 it forms the 'ghostly' part of the message for authentication purposes. When authenticating the response the terminal can also form AP and concatenate it with the received message to apply the cryptographic process of authentication.

The transaction key method has a number of variants, one of which is incorporated in an Australian national standard.[8]

Initialization of the key register can be done in a number of ways. If the problem is sidestepped by using a known constant for the initial value, assuming that all the cards have unpredictable data in their discretionary field, it will soon become unlikely that anyone has tracked the value of KR. But it is better to use an unpredictable starting value and this can be provided by having the first transaction (or a few transactions) performed with cards that are carefully preserved from illicit use. This could be done by an installation engineer or a representative of the retailer. If synchronism between KR values at the terminal and acquirer is ever lost and must be restored, a known constant must be placed in KR and the initial transaction repeated, so the best plan is that the retailer should perform this ritual. There are other more elaborate initialization methods but the ones described will usually be adequate.

Suppose that the transaction is approved by the card issuer and the response is correctly produced by the acquirer but due to a fault, perhaps in communication, the terminal cannot complete the authentication and regards the transaction as invalid. In this case the terminal will not update KR but the acquirer will and there will be a loss of synchronism. To avoid this, whenever the acquirer receives a request which does not authenticate, it tries the authentication again but with the KR value it would have used if the previous transaction had not completed. Therefore the acquirer must maintain not only the fully updated value of KR but also the previous value. With this protocol, successful receipt of a request implicitly confirms that the previous transaction was completed at the terminal. This is not essential and transaction key systems may use a confirmation message instead.

The transaction key method is an effective way to avoid the risks of tracking backwards or forwards. The most important protection is against backwards tracking because this is technically much easier for the attacker.

Derived unique key per transaction method

An alternative type of transaction key has been developed in the USA with the ungainly title 'derived unique key per transaction'. As in the transaction key method, the keys employed for enciphering the PIN and authenticating the messages change at each transaction but they change in a fixed way which is independent of the transaction data. By careful design this can provide protection against tracking backwards but not, of course, against forward tracking. For many purposes protection against the greater risk of backwards tracking may be considered adequate. The method is conceptually simpler than transaction key

and simplifies the problem of synchronization, but does not provide such good protection.

The derived key method, as we shall call it, provides a unique key per transaction for encipherment of the PIN between the terminal and the acquirer. A key derived from this one can also be used for authenticating the messages. We shall simply describe how the single key is calculated. For PIN encipherment and message authentication between the acquirer and card issuer, a normal kind of session key, often called a *zone key* is used. The cryptographic processes in the acquirer will re-encipher the PIN and must have a good level of logical and physical protection.

When the terminal is initialized, it is loaded with an initial key I and the same key is registered at the acquirer. There is an identifier to enable the acquirer to retrieve this key whenever a transaction arrives from the terminal. The value of I determines the whole sequence of subsequent keys. There are procedures in both the terminal and the acquirer for generating the derived key at any stage. The procedure can generate only a finite number of keys but this number, which is $2^{20}-1$ (equal to 1 048 575), is considered large enough for the probable life of most terminals. If necessary, the terminal can be reinitialized with the new I value.

At the terminal, the key derivation process takes place in a PIN pad and is sequential in nature. It uses one-way functions in a manner which does not allow previous values to be deduced, even if the secure environment is broken into and all the data extracted. It cannot therefore derive the key each time from I but must be able to produce the next key from stored data and always destroy sufficient previous information to avoid backtracking.

At the acquirer a different process takes place in a security module and here the starting value is always I and the derived key must be produced with a reasonable amount of computation given only the sequence number of the key. To avoid synchronizing problems, this sequence number is sent with every request message.

We shall first describe the rule by which all the keys are generated and it will then be easy to see how the security module at the acquirer can produce any one of the keys on demand with a modest amount of computation. Next we shall describe the different procedure used at the PIN pad which stores just sufficient intermediate values to enable the next key to be deduced.

If there had been no requirement for rapid production at the acquirer, a very simple process could have been used, such as iterating a suitable one-way function. This easily produces a sequence which cannot be tracked backwards but generating the thousandth key at the acquirer would need one thousand steps, which is not practical. Instead of using the one-way functions in a linear sequence, they are used in a tree structure so that the number of steps required to reach the end of the tree is much less.

Figure 10.18 shows the tree structure with the initial key at the top. From each node there are two downward paths of which the left one has no processing, it simply produces a copy of the key above. The right path in each case has a one-way function and this function is different at each level of the tree so that the results at the lowest level form a sequence of distinct keys which can be reconstructed quickly, given I. In practice, these keys are taken from level 21 but not all of them are used.

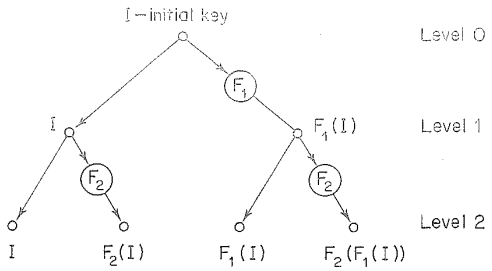

Fig. 10.18 Generation of sequence of keys

To generate a key with sequence number S it is sufficient to express S in binary form and, starting at the most significant end, use the binary pattern to decide whether or not to apply a one-way function. If the digit is 1, the function is applied. In this way, a key can be generated with at most 21 operations of the one-way function. The one-way function used is derived from the Data Encryption Standard and a fast chip can be employed in the security module at the acquirer.

To get even more rapid operation, the sequence of values S is limited to those which have no more than ten 1s. This more than halves the number of operations required and the number of keys is approximately halved except that, since I is not used, there are actually $2^{20}-1$ keys in the sequence.

A special rule is needed to increment S in this way. We shall introduce two functions which will be useful in describing later the program at the terminal. Let COUNT(S) be the number of '1s' in the binary representation of S and LOWBIT(S) be the position of the least significant '1' bit in S where the most significant bit is counted as 1 and the least significant bit as 21. By convention, if S = 0 then LOWBIT(S) = 0.

The rule for counting such that S never exceeds COUNT(S) = 10 is:

IF COUNT(S) IS LESS THAN 10 THEN S = S + 1.
ELSE S = S + $2^{21-\text{LOWBIT(S)}}$

Using this rule, if the weight of S has already reached 10, 1 is added into the place of its least significant 1 bit and therefore the weight will either stay constant or decrease, otherwise 1 is added to the least significant place in the usual way.

The method by which the current key of the sequence is derived at the acquirer is obvious from the structure of the tree. At the PIN pad, the procedure depends on a number of stored intermediate values each corresponding to a node in the 21-level tree. There is at most one stored intermediate value for each level of the tree and these form an array K(X) where X is the level ranging between 0 and 21. There are two such pointers in the algorithm, CP which points to the current key and SP which is a moving pointer used in preparing the stored array for the next key.

The algorithm is shown in Figure 10.19 where the key is derived as K(L) in the second line. Briefly it can be described as follows:

If S is odd, the required key is already stored in the 21st level as K(21) and, after retrieving it, little more has to be done except deleting the key from the array.

```
    L = LOWBIT(S)
    K = K(L) Remark: K is the current key
    IF L = 21 THEN GOTO 'X' Remark: add value of S
    IF COUNT(S) = 10 THEN S = S + 2^(21-L): GOTO 'Y'
    FOR P = L + 1 TO 21
    K(P) = F_P (K(L))
    NEXT
X:  S = S + 1
Y:  CLEAR STORED VALUE K(L)
    IF S = 2^21 THEN DISABLE PAD
    END
```

Fig. 10.19 Generation of the key at the PIN pad

If S is even, then L = LOWBIT(S) gives the level at which the key is stored. This corresponds to tracing the key from the twig of the tree at level 21 upwards, using only paths with no one-way function in them, until the stored value is found at a higher level. After retrieving this key and deleting the stored values, the stored values at all the lower levels must be recalculated using the pointer P.

An improvement to the derived key method

We know that forward tracking is possible if the stored information can be obtained from the PIN pad. With the derived key method as described, knowledge of one key of the sequence reveals some of the subsequent keys, all those corresponding to S values obtained by inserting extra 1s in the binary representation of S. The worst case is the key for $S = 2^{20}$ from which all subsequent keys can be derived. In practice, the value of the key can only be obtained by cryptanalysis but the value of some of these keys to the cryptanalyst is such that this attack might become tempting. It can easily be avoided by a small change, introducing a one-way function into the left-hand branch at the lowest level of the tree. Fortunately, this never increases the number of steps in the calculation at the acquirer. In place of line 2 of the program the following two lines are introduced:

```
IF (L IS LESS THAN 21 AND COUNT(S) IS LESS THAN 10)
THEN K = F_X(K(L)) ELSE K = K(L)
```

The extra one-way function at the bottom of the tree is shown as F_X. With this improvement it is possible to employ K(0) as the first key since it equals $F_X(I)$ and does not reveal I. Then the initialization of the PIN pad is simply to set S = 0 and K(0) = I. Without this improvement, when the terminal is initialized, the updating shown in Figure 10.19 must be performed before any keys are used.

EFT-POS with public key cryptography

The central problem of EFT at the point of sale is the vulnerability of the terminal. The two attacks we have discussed are those in which the data contained in the terminal have been extracted and the attacker either tracks backwards to reveal PINs for earlier transactions he has recorded or puts the terminal back into service and obtains PINs for future transactions which will be recorded. In favourable circumstances, both kinds of attack are prevented by the transaction key method

and backward tracking is prevented by the derived key method. If public key cryptography is used to protect the PIN, this inherently prevents back tracking and forward tracking. When the terminal has been broken into, only the public key used for encipherment is revealed and this gives no help with decipherment. Therefore, public key cryptography is a neat solution to the problem of terminal vulnerability.

A detailed design for using public key cryptography in EFT-POS has been worked out for the UK national system. We shall describe this briefly though not with details of key management at higher levels. Figure 10.20 shows the secret and public keys held by the main entities which take part and the relationship between them. For example, sk_T is the secret key held in the terminal and pk_T is the corresponding public key held by the acquirer. These keys are generated by the terminal during its initialization. Similarly the acquirer, card issuer and key registry each generate a pair of keys and make the public key known to others, as shown in the figure. The arrows point from secret key to public key and indicate how a signature can usefully pass from one entity to another. Such signatures actually accompany the request messages from terminal to acquirer to card issuer and the response messages in the opposite direction as the diagram shows.

Each entity, when it generates its keys, must register the public key with the key registry — a body which functions for the whole system. The key registry has its own secret key sk_R and the public key pk_R is made known, with careful precautions for its authenticity, to the terminal, acquirer and card issuer. When any of these entities requires an authentic public key it is received in a message from the card registry containing the key, the identifying data and a digital signature using the registry's key. Before using any public key for the first time, each entity checks its signature in this *certificate* which it has received. Terminals receive their certificates from the acquirer when they need them.

Until about 1987, it was difficult to find equipment to carry out the public key cryptography fast enough. By using signal processors, such as the Texas Instruments TMS320 with suitable software, a low cost solution for operation at reasonable speed in the terminal has become available. At the same time a specialized RSA chip has appeared which can perform these functions fast enough to meet the requirements of the acquirer, card issuer and key registry. With a short public exponent (65537 was used) the operations of encipherment and

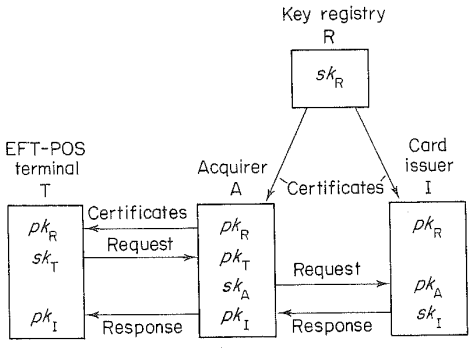

Fig. 10.20 Public key relationships for EFT-POS

signature checking are made very fast. Signature employs a one-way function using an RSA operation with a public key distributed by the key registry for which no secret key has been retained when it was generated.

With these keys in place, the security services for the request and response messages are as follows.

The PIN block is enciphered using the public key of the card issuer. There are many card issuers, but a small number of these issue the majority of cards and the terminal holds a cache of authenticated public key values for this purpose. If a card arrived for which no public key is held, the terminal requests this public key from the acquirer, which holds a complete set and provides them as certificates. Checking the certificate at the terminal is a rapid process. PIN block encipherment is provided from end to end, terminal to card issuer and this meets one of the main security requirements in a simple way, without depending on human operations to handle secret keys at the acquirer.

Authentication of messages is performed by the digital signature technique which simplifies key distribution since secret keys remain where they have been generated. The request from the terminal is signed with sk_T and this signature is checked at the acquirer which holds the corresponding public key. It would be nice to check the same signature at the card issuer, but that would complicate key management because card issuers would have to hold public keys for all terminals. Instead, the message is re-signed by the acquirer using its own secret key for which the card issuer holds the corresponding public key pk_A. Re-signature at the acquirer places some trust on the acquirer's software but authenticity of the request can, to some extent, be covered by the authenticity of the response. The response is signed by the card issuer with its secret key and this response can be checked by the acquirer and also passed to the terminal for checking. Provided the essential information on which the transaction depends is repeated in the response, the necessary trust in the acquirer re-signing the request message can be largely eliminated. The advantage is that both PIN block encipherment and transaction authenticity are virtually end to end.

Digital signature has another bonus. If an event is recorded in the audit trail at each entity as a signed message, and assuming these have sequence numbers to prevent insertion, deletion and copying, they can subsequently be checked by an auditor using the public key and they provide non-repudiation in its purest sense.

In practice, an EFT-POS scheme using public key cryptography may have to differ from this ideal in a few respects, but the description of the principle shows that all the requirements can be met in a very complete and simple way using public key cryptography. It is possible to arrange for all the key pairs to be replaced from time to time, if the users require it, except that the terminal's secret key would remain unchanged during the life of the terminal or, if it has to be changed, the terminal would be re-registered and treated as a new terminal for purposes of initialization. Terminal initialization involves registering the public key and storing in the terminal the public key of the key registry. The easiest method is to bring the terminals to a secure area under the control of the key registry for this purpose, though registration schemes based on the principles described by Matyas can be used.

In order to perform the signature reliably and handle the full level of traffic,

acquirers and card issuers need more than one security module capable of performing all the cryptographic functions required. These security modules must share the secret key and an exchange of secret keys between these modules requires careful design to minimize its dependence on the operators and provide the necessary dual control. In a similar way, the registration of public keys requires careful operational design and similar considerations of dual control. Standardization of key management for EFT-POS will probably cover all the details of the messages exchanged and some aspects of initialization, but not the higher levels of key management since a degree of confidentiality here helps to keep systems secure.

As soon as it is freed from the constraints of processing power, public key cryptography in this context provides a simpler solution than symmetric algorithms. There is some additional message overhead but, with ingenuity, this is reduced to little more than is needed to make up the units of 64 bytes employed in RSA cryptography.

Off-line point-of-sale terminals and smart cards

Since an off-line terminal cannot contain a complete list of PIN values against card identities, it must apparently employ an algorithmic relationship between the PIN and data contained on the card. Since card issuers do not wish to share PIN information they must use different keys for their PIN algorithm. Off-line point-of-sale terminals which can accept cards from many issuers would need to hold the keys of all those issuers. Since point-of-sale terminals are physically insecure it is unlikely that card issuers would share their secret keys in this way.

Public key systems cannot help, though there have been suggestions that this is so. We can devise an effective way to use a public key system to check the PIN against data contained on a card. For example, if P is the PIN, I the card identity (account number, issuer, card's serial number) and R a reference value stored on the card, the value of R can be generated by using the secret key sk of an RSA cipher as

$$R + P = (P + I)^{sk}.$$

Any method of addition can be used. For checking the PIN, using the public key pk the relationship used is

$$P + I = (R + P)^{pk}.$$

This does not allow P to be obtained from card information because P is on both sides of the equation.

Unfortunately, this scheme is vulnerable to a PIN search because all the information needed to check a PIN is available and the searching can be carried out by a fast process without using a terminal. No process which uses only public information to check a PIN can be secure against PIN searching, unless the range of the PIN is sufficiently large, which probably means ten digits or more and is not acceptable for payment systems used by the public.

The remedy seems to lie in the use of tokens which contain processing and storage ability, the so-called 'smart cards' which we call *intelligent tokens*. These contain a microprocessor and a data store which does not need electrical power

to maintain it. When the token makes connection with its terminal, the microprocessor and store access circuits are provided with power and the token is able to apply its intelligence in order to protect data in its store and control access according to pre-arranged rules.

Physical security requirements of the intelligent token

If it were possible to open up the token, read all the data within it and then make a similar token containing these data, the token would have very little advantage over a magnetic-striped card. It gains its advantage either by preventing the reading of its data or making it exceptionally difficult for the enemy to manufacture cards with the required properties. This difficulty of manufacture is a consequence of making the token in the form of a thin plastic card, so that readily available electronic components cannot be used to counterfeit it. It would seem a very difficult, though perhaps not impossible, task to read the data from a card. The effort required would only be worthwhile if it gave a large pay-off in the form of duplicate cards which could be exploited for fraud.

PIN checking in an intelligent token

By checking the PIN within the token itself, it is possible to use an arbitrary PIN value. There is no key for checking an algorithmic relationship and the card issuer need share no secrets with the acquirer's terminal. The principle is simply that the PIN value is entered on a keyboard in the terminal, transferred directly to the card through its electrical interface and the checking of the PIN is carried out within the token. The token then signals to the terminal if the PIN is correct. The token itself can enforce rules to prevent PIN searching. It can also apply rules on behalf of the issuer to limit the amount of payment the card may make within a certain period. In doing this, the card depends on a current date given it by the terminal. Falsifying this date in the terminal produces transaction records which are unacceptable to the issuing bank and is unprofitable to the merchant.

For example, the issuer could impose a limit on the total payment provided by the token so that the customer requires a new token to continue making payments. The issuer could limit the amount of payment in any single transaction or series of transactions to the same account, or limit the total number or amount of transactions in one day. Any of these rules can be applied within the token just as well as by an on-line system. Only the absence of a comprehensive and up-to-date blacklist makes off-line operation less secure in this respect.

The matching of the PIN within the token could use a reference PIN value. Alternatively, a one-way function could be used, which does not require the PIN to be stored and this might seem to be more secure, for the PIN cannot be read from the token (even if the reading technology is available to the enemy). However, this approach is not really more secure because the small range of the PIN allows it to be found by searching whenever information is available about how the checking takes place.

The result of a successful PIN check may be to generate a properly authenticated request message for sending to the card issuer. In an on-line system this results in an authenticated approval message telling the merchant that the payment has

been made. In an off-line system there is a further difficult requirement, which is to assure the merchant that the PIN checking has been properly carried out and that the match is genuine. Even without a micro-processor it would not be difficult to construct simple logic which always gave the 'PIN match' response to the terminal, whatever PIN was inserted. Verifying the PIN within the token is easy but conveying this message convincingly to the terminal can be difficult.

The use of a one-way function would, it seems, allow the terminal to check the PIN it has been given with the PIN held in the token without the ability to read the PIN from the token. For example, the terminal generates a random number x which it sends to the token and receives in reply the response $EP(x)$, where x has been enciphered using for the key a quantity derived from the reference PIN, P. By performing the same calculation with the PIN keys in by the customer, the terminal is able to verify the PIN without being able to discover the reference value in the token. In this way the terminal verifies that the token actually contains the pin and is 'intelligent'.

Unfortunately, this scheme fails because of the small range of PIN values, which allows PIN searching. Figure 10.21 shows an alternative procedure which partially solves this problem and allows both the terminal and the card to verify that the entered PIN is correct. In place of the encipherment function E we have used here a one-way function O such that, if

$$y = O(k, x),$$

a knowledge of values of y and x does not allow k to be determined. We can think of k as having at least 56 bits to prevent searching. The function O could be replaced by encipherment with the DES algorithm, but an encipherment function is not essential and the one-way function $O(k, x)$ can serve just as well.

In the procedure of Figure 10.21, the PIN, P, which can be thought of as a 16-bit number, is concatenated with a 40-bit number C to form a 56-bit 'key' for the one-way function. The terminal sends the value x to the token which returns the response

$$y = O(P||C, x).$$

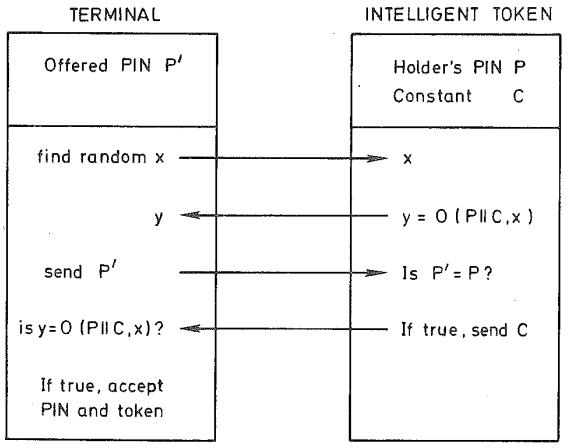

Fig. 10.21 Protocol for intelligent token authentication

There is no way in which the reference value P could be obtained by searching unless C is known. After this response, the terminal transfers to the token the trial value, P', which has been entered by the customer. The token's processor checks this against its reference value P. If the two match, the token returns the 40-bit number C and the terminal then computes $O(P||C, x)$ and checks whether this agrees with the value y received earlier.

A token made by an enemy without a knowledge of P would be unable to give the response y. Because y and x have 64 bits there is probably only one value of $P||C$ which matches the x, y pair and this contains the 16 bits of P in their correct places. There is no way in which the token could find a value of C to satisfy the final check in the terminal unless it contained the value P at the time of its last response.

Thus the token is able to check that the correct value of P was entered at the terminal and the terminal is able to check that the token contained this value, assuming that the PIN was correctly entered.

Having once obtained the value C, the token is an 'open book' to the terminal but this is no loss since the terminal could equally well store the value of P associated with a successful match. The value C is of no greater value to it than P.

Following the correct match of the PIN, the card would be prepared to generate an authenticator so that the transaction record sent by the acquirer to the card issuer will be accepted as a valid authority for payment. In effect, this functions like a bank cheque and is an instruction to the issuer to pay a specified amount into the account of the acquirer. No PIN need be carried, because the valid authenticator is a sign that a correct PIN was entered.

The authenticator for this message employs a key which is stored in the token and is known only to the card issuer. This key is specific to the customer and when the payment is processed by the issuer, the customer identity in the message will allow his authentication key to be extracted from a table and used to check the authenticity of the entry.

We can now see clearly the advantage of the intelligent token. Its intelligence is used on behalf of the customer and the card issuer in the potentially 'alien' environment of the point-of-sale terminal. The token checks the PIN to verify who is presenting it, then authenticates a message to its 'master' which is the card issuer's computer system. From the issuer's viewpoint this is secure. The authentication key is customer specific, so there is no large gain from extracting the key from a token.

The acquirer is in a less favourable position because his terminal cannot verify the authenticator value. The method we descibed above (see Figure 10.21) goes some way to authenticating the token but cannot verify the end result — a message which is the authority for the card issuer to make the payment. In the off-line situation the merchant may be vulnerable, depending on what kind of indemnity the card issuer gives.

To help the acquirer it has been proposed that the token add a second authenticator using a key known to all terminals, which his terminal can check. But such a key contained in all terminals and all tokens is really too exposed.

Used in on-line systems, smart cards greatly improve security compared with the easily copied magnetic stripe. Off-line, they also improve security, but leave the residual weakness we have described. Public key cryptography can help.

10.6. PAYMENTS BY SIGNED MESSAGES

Our discussion of payment systems has emphasized the value of on-line connection to bank processors which enables card validity to be checked, the use of stolen cards detected and the account of the customer examined before the payment is authorized. Today, many payments are made without reference to a bank, for example by means of a cheque. Either a bank has been willing to guarantee the payment up to a certain limit or the beneficiary must trust the drawer of the cheque.

There will in future be many occasions when payment through a communication network is useful but reference to a bank is not convenient. For example, an on-line information service charges a fee but a caller does not have to be a known subscriber to the service; he can be anyone who 'presents a bank card'. The need for this method of payment, an 'open shop in information services' was pointed out in a note by Raubold.[9] It is the electronic equivalent of a bank cheque, which requires some trust between the parties. In this kind of transaction, the communication path betweeen the two parties is already established as the means by which the information service was provided. Payment for tickets for transport and entertainment can follow the same pattern.

The implementation of an electronic cheque requires technically little more than a digital signature facility with a key registry to authenticate public keys. A hierarchy of keys is proposed in which a registry authenticates the bank's public key and the bank authenticates the customer's public key. In order to allow the content of the cheque to be validated with a minimum of data beyond the cheque itself it should contain all the items listed in Figure 10.22. These items fall into three sections. The first is, in effect, a certificate by the key registry which authenticates the bank's public key. It also includes an expiry date so that the bank need not worry that an old signature is being used after it has expired. Anyone can verify the bank's public key using the signature and the public key of the key registry, which is available in many different places to avoid falsification. The key registry could be the central bank of a country, for example.

The second section of the cheque contains the customer identity and his public key, signed by the bank and therefore verifiable using the public key stated in the first section. This again has an expiry date to prevent its indefinite use. These first two sections of the cheque are constants which appear on every cheque drawn

1 Bank identity	2 Bank public key
3 Expiry date	4 Signature of 1–3 by key registry
5 Customer identity	6 Customer public key
7 Expiry date	8 Signature of 5–7 by Bank
9 Cheque sequence number	10 Transaction type
11 Amount of payment	12 Currency
13 Payee identity	14 Description of payment
15 Date and time	16 Signature of 9–15 by customer

Fig. 10.22 Electronic cheque

by this customer during their period of currency. They are followed by the payment information which forms the third section of the cheque data.

A sequence number ensures that each cheque is different from any other and provides a method for detecting duplicates. The transaction type is provided in order to use this same format for other purposes, described later. The amount and currency of the payment and identity of payee describe the payment to be made and there is a field for 'description of payment' which helps the parties to associate the payment with its purpose in their records.

Because electronic cheques can easily be copied, the drawer of cheques or his bank should examine the account to look for possible duplicates. Items 9 and 15 of the cheque allow them to be sorted into a sequence in which duplicates would immediately show up. It seems more practical for the bank to eliminate duplicates, and this is simplified if there is a time limit for the presentation of cheques for payment.

The final signature, by the customer, covers all the variable information in the cheque. In order to form this signature the customer needs a secret key which he must protect against disclosure.

A corporate customer of the bank will protect his secret key in a physically strong enclosure. Access to this signature can be controlled by physical keys, passwords and tokens, with strict limits on the extent to which it may be used by each individual. Private customers of the bank can carry their keys more conveniently, for example in an intelligent token or 'smart card'. A necessary protection against misuse of a lost token is the conventional PIN but perhaps the amount at risk justifies using a dynamic signature or some other personal characteristic to protect the digital signature against misuse. A terminal is needed in order to generate a cheque, sign it with the aid of the token and send it to the beneficiary. This can be any intelligent terminal which has the interface for connecting the token and a PIN keyboard or a pad for a dynamic (manual) signature, whichever of these is used to protect the token. It will often be the same terminal to which the service has been received for which the payment is being made, for example a theatre ticket issuing machine.

Functioning as an electronic chequebook, the private customer's token can record the transactions it makes and list them for its holders at any convenient terminal. The smart card used for point-of-sale payment has been called an 'electronic wallet'. The cheque is more versatile because it can be sent to anyone who has a means of recording the data and presenting them at a bank. No secrets are present in the terminal. This makes possible 'home banking' with reasonable security. If an account balance is requested, for example, the request is signed (verifying that a correct PIN was used) and the response is enciphered by the bank with the holder's public key. Only the correct token with its secret key can decipher the response.

Having personalized the token for its customer, the bank need keep no record of the secret key. New tokens use new keys and the bank's record of public keys is updated. PIN storage is part of the personalization of the token at the start of its life. The bank needs a record of the PIN only if it wants to remind the customer who has forgotten it. Customers can choose their own PINs but it is not a good idea to allow the PIN to be changed because this introduces an extra risk.

The date and time item in the cheque format depends on obtaining a reliable time value from the terminal. In principle, a date and time service could be provided on the public network with its values signed by the key registry, but this spoils the simplicity of off-line usage. The time stamp may therefore be optional. For corporate cheques, time might be used regularly, to provide timed evidence of payment and to help in logging the use of the cheque-writing facility.

The first motivation for the electronic cheque was an off-line payment for an information service. The service provider collects the cheques and presents them in batches to its bank for payment. They can be sorted and cleared in a conventional cheque-clearing operation. Cheques necessarily involve some trust that the account on which they are drawn is capable of payment. The same electronic cheque could be presented for payment as soon as it is received, using an on-line system, and thus cleared. This is an on-line payment mechanism which verifies the payment immediately. In its appearances to the customers it is no longer a cheque but a point-of-sale payment.

Point-of-sale payments by electronic cheque

The mechanism that we have proposed for a cheque enables an off-line payment to be made to a merchant's terminal. The terminal collects the cheques, comparing them if necessary with a local blacklist, and presents these cheques to its bank in a convenient batch. The bank sorts the entries and sends them to the respective card issuers for payment. The merchant's terminal can verify the signatures and the merchant is sure that the cheques will be accepted as genuine.

For on-line operation, the electronic cheque is transmitted at once to the card issuer bank where the signature is checked and the blacklist and drawer's account examined. If all is well, a payment message, signed by the issuer bank, is sent to the acquirer bank. When this has been checked, a signed advice goes to the point-of-sale terminal. If necessary, the accounts of the customer and merchant can be updated at once and the acquirer bank can issue a signed receipt. These are matters which depend on the precise payment rules, which must be decided when banks introduce new payment services.

By using public key signatures in this way, the security of each stage of the payment process is made dependent on the care exercised by the person or organization that would lose by fraud. Nobody is dependent on another's security. Thus each party is responsible for the security of his secret key and controlling its use. At the end of the payment process, each of the parties holds signed messages which record the commitments of the others. The merchant holds a copy of the cheque and the acquirer bank holds a copy of the payment.

If digital signatures have a weakness, it is the fact that signed documents can be copied and this places on the issuer bank the requirement to detect and eliminate duplicate cheques. The cheque sequence number is administered by the intelligent token and is therefore as secure as the customer's secret key.

The same intelligent token which provides an electronic cheque between individuals can therefore function for off-line and on-line point-of-sale payments. It can also 'cash a cheque' at an ATM with on-line verification.

Figure 10.23 shows how this works in a shared ATM network. The customer's request is formed into a message and presented to the token. Here it is signed

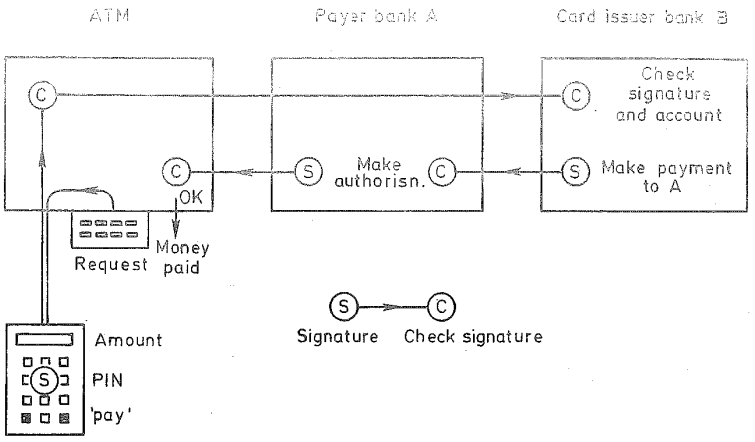

Fig. 10.23 Shared ATM network using digital signatures

and returned to the ATM. The ATM checks the signature, to avoid passing ineffective messages into the system. If it is correct, the 'cheque' passes via the payer bank, A, to the card issuer bank, B. Here the signature is checked and the customer's account examined and, if everything is in order, debited. A payment message signed by B is sent to A. The message and its signature are checked and if all is well an authorization goes to the ATM to release the money. The signed messages form an audit trail. All the messages carry similar information but their purposes are different. To differentiate these messages and provide for even wider use, the format of Figure 10.22 includes the *transaction type* which denotes a customer cheque, ATM request, interbank payment, payment advice and so forth.

If public key signatures come into use first in payment systems, the financial institutions will establish key registries. It would be natural for a verified public key issued by a bank to be accepted as a kind of bank reference. In this way the operation of key registries may become a function of financial institutions.

A development of the intelligent token

The 'smart card' in its current form depends on a separate terminal for its input and output. This is a security risk in several ways. The PIN enters the card from a keyboard, which could be used by an enemy to capture PIN values. The 'cheque' enters the card from the terminal and the card's owner cannot be entirely sure what is being signed. In an on-line payment, the message coming back from the centre is displayed by the terminal and could be falsified. For example it could indicate that a transaction has been cancelled, a refund has been made or a payment refused, when this is not true.

To avoid these serious risks, almost the whole of the signature operation should be under the control of the owner of the secret key. The PIN keyboard should be part of the token, so that the PIN is not entrusted to a 'foreign' device. When a payment is made, the most critical data, which are the amount and currency, should be shown to the payer by a display which he can trust. For this purpose

a display is provided on the token. The display also shows any messages the token receives that have been signed by its own bank, such as an advice of a cancelled payment or refund. It is only by placing these vital functions in the token that their security is maximized. Figure 10.23 shows an ATM network using a token of this kind.

There are other functions of the token that are less critical but convenient. It can maintain a local balance of its account which errs on the side of caution, since received amounts will not be credited until a suitable on-line transaction is made. Automatic teller machines can provide an update, but to provide this for 'foreign' cards needs a little more organization and, possibly, a small charge. The account information is easily protected cryptographically, using the token's secret key for decipherment. Since this message concerns nobody else, its integrity can be determined by a secret password held in the token and known to the bank.

Such a token becomes a pocket terminal, capable of coupling to a point-of-sale terminal, an ATM or to a home computer or videotex terminal for home banking. It seems unlikely that home banking can be secure without such a 'sealed' device holding the key to authenticate messages. The coupling can be optical or magnetic, to avoid the nuisance of contact problems. It should have a battery to make the secret key more secure by overwriting it if a physical attack or a PIN search is detected. The extent of physical security is necessarily limited in a small device. This makes it useful to add a time and date stamp to transaction messages so that misuse can be tracked after a token has been reported lost by its owner.

There is no inherent reason why a signature token with its keyboard and display need be thicker than a standard bank card. Calculators of bank card thickness are on the market. But for some time the tokens may have to be thicker than a card to enclose the RSA chip, the battery and perhaps some extra physical protection for the secret key. The customer will expect an advantage to compensate for the increased size. If the token, while limited to a single account, works for point-of-sale debit and credit transactions, ATMs and home banking, this is already a help. In principle a signature token could operate with more than one account, holding the corresponding secret keys and taking its cue for selecting the account either from the terminal it connects with or from its keyboard. Therefore, one token could replace many or all of the cards we now carry. The different signatures can have their own expiry dates. Assuming that its secret key does not have to be altered, a signature facility can be renewed in several ways, either by a number through the post, at an ATM or at a bank branch. The big question is whether financial institutions that are in competition would be willing to cooperate to this extent. The cost of a signature token that is well used can easily be met from the saving in operating the payment mechanisms.

Almost all the technology needed for the 'signature token' is available but it must be put together into one well-engineered packaged. In addition to the technological development, signature and message format standards must be agreed.

10:7. ACCESS CONTROL BY INTELLIGENT TOKENS

We have proposed an intelligent 'signature token' in which the authenticity of the public key associated with the signature is certified by a bank. Other organiza-

tions could provide the certification and the key registries but the banking community is probably the best choice for the system to be widely accepted in most countries. When such a system is in place and working it can be used as a means of personal identification for purposes additional to banking. We will look at two possibilities, access control for buildings and access control for distributed information systems. Both applications will need little more than the ability of the token to sign a message. To the banking system, this kind of message has no particular meaning but it must be distinguished from a payment so that the signer cannot be fooled into making a payment. If the 'transaction type' signifies that a payment is being signed, the token's display must show this and display the amount.

A person who is to be granted access to a building must go to an official for his token to be recognized. This procedure inevitably requires an independent method of verifying the person's identity and his right to access. For introduction to the access system, the token is placed in a reader and is given a short test message to be signed. The access control system has then copied the token's public key and can check the signature to verify that the applicant really owns the token (more accurately, that he knows the PIN). The public key is stored, together with the associated data such as the rights and limitations of access and with a local identifier sufficient to allow the access rights to be cancelled when necessary.

To enter the building the token is presented to a terminal, which gives it a message to be signed. If the signature agrees with the *stored* public key, the token and its holder are assumed to possess the stored access rights. The access message must take a standard form so that it cannot be misused for other purposes. In principle, the transaction type 'access' together with a date and time are sufficient. An access control system could fraudulently present the token with a message timed for later, then use that signed message in a bogus token to gain access somewhere else by impersonation. The way to avoid this is to present on the display the name of the building or the authority controlling the building. The signer should always check what is displayed before giving the token his PIN.

By making one token take on the many roles which are presently given to several cards, passes, badges etc., the risk due to a lost token is increased, but cancellation of lost tokens can be made easier. Banks could provide as a service the daily listing of cancelled public keys.

Access control for centralized and distributed information services

For a centralized information service the method of access control can use the same principles that we described for access to a building. The introduction of a new user requires that his right of access be known. If the service is for sale, that right is conferred as a result of a payment. An electronic cheque (cleared if necessary) not only provides the money but also gives the service provider its reference data in the form of a public key.

Subsequent access control uses a different message on each occasion, because of the date and time field, so it is protected against the replay attack. The signed message can be held by the service provider as evidence if there is a dispute about illegal access. But the 'signing on' message does not protect against an enemy who takes over the channel subsequently. At the present time, public key ciphers

are too slow for use to protect the channel. If the DES (or a similar cipher) is used, the server can provide a key enciphered under the user's public key and the token can decipher it, using the secret key.

The user's access rights are stored in association with his public key and are put into operation whenever a correctly timed message is received with a signature which matches. For very sensitive or valuable information it is possible to check the public key with the bank which issued it — the fee for this service would be similar to the fee for rapid clearance of a cheque. This scheme of access control is appropriate for a centralized system in which the various users' access rights tend to be independently controlled. If there is a lot of sharing of access with frequent granting of access from one user to another, a more flexible method is needed.

For a distributed information system, access control by *capabilities* is more flexible[10] and it allows sharing of data and procedures to be managed by the users themselves.

A static model of the access rights in an information system is provided by the 'access matrix' in which are stored the access rights of each 'subject' relative to each 'object'. A *subject* can be a person (system user) or a process created by a person or created by another process. By subdividing in this way, the rights of individual processes can be very limited indeed and this prevents the 'Trojan Horse' attack. An *object* can be data to be read or written or a procedure to be invoked and entered. In some systems there are other objects like input/output channels or queues. The entries of the access matrix have a correspondingly wide choice of rights such as read, write, append, enter (subroutine), join (queue) etc.

The method of access control implied by the centralized access control we described earlier amounts to storing, for each object, a table of subjects and their relevant entries in the matrix. On the other hand the *capability* method provides the matrix entries themselves as 'tickets' to be held by the subject and presented when they are needed. Checking their validity is easy. Like a doorman in a theatre the controller looks only at the ticket and is not concerned with identifying the subject. But this carries the requirement that tickets or 'capabilities' must not be forged or copied.

In monolithic systems capabilities are handled in a special way, sometimes with special hardware features such as a 'type' bit in every word, and the user may store and retrieve his capabilities but not create new ones or copy them except with the help of the operating system. In a distributed system the user needs to keep his capabilities entirely under his control, for example in an intelligent work station, in order to present them at any suitable server.

In a distributed information system we would like files of data to be mobile, able to transfer to any convenient file server, and also that procedures can be found in alternative places. Users should be able to share with others their access to data which they 'own' or to their own procedures. But if user Ann gives Bill access to a file she may want to prevent Bill giving the access to Charles without her consent.

These requirements are reflected in the way that capabilities are formed and protected. Each object can have its own scheme for coding and checking the capabilities which give access to it but others should not be able to construct these capabilities. For this purpose the object (or a group of objects) has a secret key

which is used to encipher all capabilities for distribution. In this way the capabilities can be assembled and marshalled by users and, to help this, the nature of the capability (the object and means of access) can be associated in clear form, but the 'doorman' looks at the part that was enciphered.

This encipherment does not prevent copying, it only prevents forging new values. We do not want to prevent Ann copying some capabilities to Bill but when Bill presents them we need to know of Ann's intention and permission. It is clear that Ann's capabilities must contain Ann's identifier and a flag to signify ownership. Bill's copy can be coded to prevent further copying.

If these capabilities are to be valid at all those servers where the files or procedures are available, using conventional methods each server would need to recognize Ann or Bill by their passwords, but it is unwise to replicate secret data in this way. Public key signatures provide the appropriate mechanism. A service request from Ann must contain the capabilities needed to carry it out. Her own work station can assemble these. The request should also be signed, timed and dated by Ann. The server will act on this request if the signature matches the reference value of her public key. If necessary one of the capabilities can contain the public key. When she wants to share a capability (one that is marked for sharing) with Bill she sends it to him in a signed message meaning 'I share this with Bill but not for him to own' and giving Bill's public key. The server can check the signature and provide Bill with his copy of the capability, marked for his use but not for copying.

This method of access control allows objects to be replicated and distributed, carrying with them their 'doorman' in the form of a secret key and a method of coding capabilities. They hold no secret information about the users, only their public keys, which can be obtained from other servers. In this application, the intelligent token is required to give a signature to an access request. It may also be required to decipher a message from a server which contains a secret key to protect the communication channel. Such a message should be date and time stamped.

If the application requires the server to be authenticated, the user must know the server's public key and be able to check its signature. Potentially the intelligent token is able to do this but its ability to hold other public keys and perform the interaction needed for checking them raises the question of how much its function should be expanded. In all other matters it has been required only to perform RSA operations with its secret key.

By describing the use of the token for access control we have shown that the use of digital signatures goes beyond banking and that the signature token can serve in these other applications without complicating its function very much. The use of the system is widespread rather than specific to one organization, therefore the economic incentive to develop and install such a system will need many organizations to get together.

10.8. NEGOTIABLE DOCUMENTS

A negotiable document can be given by one person to another and confers some right or value on the person who holds it. The word 'person' includes legal persons, like companies or corporations. Thus it can be owned or sold like any other

valuable thing. A theatre ticket is negotiable when it admits not a named person but whosoever presents it at the entrance. A banknote is another example. These documents are often worth stealing so they must be kept in safe places.

There are some payment tokens that are negotiable. These are bought from a shop and used to pay for transport, telephone calls, parking and so forth — usually small payments. Some payment tokens can be used several times, until all their value is consumed. These are called *pre-payment* tokens. An example is the 'Hologyr' card which has one or two tracks of hologram impressed on a plastic surface (and covered for protection). These can be read by infra-red light and small pieces of them successively destroyed by heat. The reading machine reads the next piece of hologram, then destroys it while examining it optically during destruction to verify that it is the genuine thing. Similar tokens can be made with the Drexler material, where very small holes are fused into a metallic film embedded in the card. The security of these cards rests on the difficulty of making or obtaining the materials and constructing something which functions like the real thing.

The intelligent token can also function as a pre-payment card. It must hold some secret data which are recognizable by the reading station. The secret could be a cipher or one-way function or a secret key which takes part in a cipher, authenticator or one-way function. If the enemy can make an intelligent token or something which functions like it, he must be prevented from finding the secret, therefore each challenge to the token and its response should differ from previous ones. The secret is held in each token and in the reading station, so the physical security of both must match this level of exposure. Using public key ciphers, the reading station need not hold a secret, but since the reader can use the same physical protection as the token, or more, this is not a practical advantage. The essence of a pre-payment token's security is the difficulty of forgery, like any other negotiable document.

A general-purpose negotiable document

A signed paper document can have three types of significance, informational, evidential and symbolic. The easiest to provide is its information content, since the whole evolution of information technology has been for the purpose of handling information. But compared with a signed paper document, the ordinary electronic one has lost its *evidential* value. Changes to a paper document show, whereas a digitally stored document usually gives no reliable evidence of its origin or the alterations that may have taken place. The digital signature (chapter 9) restores this evidential significance. With the aid of key registries, digital documents can then be used as evidence of their origin and accuracy, provided that the secret keys have been handled with care.

The *symbolic* function is the one which gives significance to a document's ownership. A theatre ticket symbolizes an access right to a seat at a performance. Negotiable documents are the ones that employ this symbolic significance. Digital signatures are not enough, because a digitally signed document can be copied and nobody can distinguish the original from the copy.

When a situation seems to need a negotiable document it is sometimes possible to change the whole method of operation and avoid that need. For example,

some kinds of bank cheque are negotiable, but the 'electronic cheque' we described earlier relates only to one beneficiary account. If duplicates reach that account they are detected and only one credit is made. Negotiable documents can be handled electronically if there is a central register for all documents of a certain category, where their ownership is recorded. All changes of ownership are reported to the register and the changes acknowledged, as part of the procedure for the transfer. A central registry has been established for shares and bonds quoted on the Copenhagen Stock Exchange[11] and there is a project called INTERTANKO[12] to register the ownership during the shipping of bulk cargoes. These registries depend on trust in the central organization and on secure communication between the document holders and the centre. A general scheme to enable (original) documents to be distinguished from copies has been proposed.[13]

If a registry can be established for each category of electronic negotiable document, the symbolic aspect will never be needed. This may be the way things will go, because the infrastructure for cheap and efficient data communication is spreading rapidly. It is not necessary to store the whole document centrally since a suitable hash function or signature can represent it, but it is necessary for each registry to recognize, communicate securely with and authenticate its clients, the document holders. A registry may prove expensive to run because of these security requirements.

We will, therefore, look at an alternative, which is to establish electronic negotiable documents in which the evidential and symbolic properties are managed without registries of ownership. This scheme may prove useful, for example bills of lading are an important form of negotiable document and some countries want to have these documents retained. But even if 'electronic negotiable documents' are not found to have practical value, they make an interesting exercise in security technique.

The information document can take any form but is usually in natural language, which states the purpose of the document. If certain forms are widely used, there will possibly be standards for the wording and format, but these are not essential. There can be any number of copies of this part. The evidence is provided by a digital signature. The value of the document depends on the verification of the signature, and therefore on the availability of an authentic public key. We described earlier a way in which the banking community can provide an authentication hierarchy for cheques and this hierarchy seems entirely appropriate for all other documents used in commerce.

The symbolic part of the document must be unforgeable and contain a secure secret, the existence of which can be verified. The choice we shall make is to use the same device as the 'signature token', with its keyboard and display, except that in this application the secret key refers to the document, not to a person. The token may also need some special, application-dependent features that are described later. To link the token (the symbolic part) to the information and evidential document we use the public key. For example, the signed document could contain these words: 'This document is owned by the possessor of the token containing a secret key to match the following public key: (quote value here).' The signature on the document prevents the quoted public key being falsified and the validity of the symbol token can be checked by obtaining its signature.

A random number is sent to the token and signed. The signature is checked using the public key quoted in the document. If they agree, the token is genuine and is the one relating to the information document.

To create a negotiable document, the originator uses a secure system to generate a pair of matching keys for a signature method, such as RSA. The secret key is installed in the token and 'sealed in' by an irreversible change (like blowing a fuse inside it) so that the secret key can never be altered. At the same time the token is given enough identification to link it with the information document. There is no reason why the whole of the document's information should not also be contained in the token. The public key is quoted in the appropriate part of the information document, such as the statement quoted above, and the whole is signed. The document signature can also be loaded into the token and sealed in, so that no other document can be associated with it. After these steps have been taken, there are both information and symbolic parts to the document, each matching the other. For valuable documents the signature method should be very secure, because an enemy can use a lot of time and computing power attempting to match a fraudulent document to a symbolic token he happens to hold and for which the public key is known. Additional security is given by using only formal documents with a precisely specified wording.

We have described the creation of a document. Checking a document entails these steps:

1. Read the information to verify that it states what it should, that the originator had the authority to issue it and there are no unexplained random parts which might indicate a forgery by the 'Birthday method' (see page 279).
2. Obtain an authentic public key for the originator. This can be obtained from certified public keys in the document itself if they trace back to a well-known key, such as the public key of a national bank.
3. Verify the signature on the document, using this public key.
4. Check this signature against the one contained in the token, if there is one.
5. Send a random message to the token and receive back a signature.
6. Verify this signature with the aid of the public key quoted in the document.

If all the checks are passed, the token is assumed to be the genuine 'symbol' for the information document in question. The value of the combined information and symbol depends on what is stated in the document and on the trustworthiness of the originator.

Destruction of the symbol can be physical destruction, or it could be brought about by a command at the keyboard of the token. Some negotiable documents are not normally destroyed, others come back to their originator for destruction (like a banknote). In the latter case, the destruction can be 'logical' using a password entered by the originator at the time of creation; in this way, inadvertent or malicious logical destruction is avoided. The end result of logical destruction is to destroy the secret key held in the token.

Protection of negotiable documents against theft

A negotiable paper document must be guarded against theft. Electronic 'symbol tokens' must also be kept safely because of the inconvenience of losing them.

Protection against theft or misuse can be achieved logically, removing the incentive to break into a safe in order to steal them.

Each owner can give the token a password and then lock it against further use until the password is presented again. While the token is locked, it will not respond to a request to sign a message with its secret key, so it is useless to a thief. When a transfer is made, the vendor unlocks the token and the buyer locks it by giving it a new password of his choosing.

Since a transfer employs the intelligence of the token it is possible at the same time to enter a record of ownership and dates of transfer, so that each symbol token holds a log. This could be useful if there are suspicions of theft or misuse. Each type of application for these tokens will have its own special requirements and there is a good chance that the software of the token will allow it to help the owners in other ways. For example, supposing an owner loses the password, the document becomes inaccessible. To recover from this there can be an 'emergency password' which unlocks it. Usually, the originator is trusted by all subsequent owners. If the originator keeps the emergency password, any subsequent owner who loses the current password can bring it back to the originator to be unlocked.

The need for electronic negotiable documents is still uncertain, but the possibility of making one (in the way we have described) shows that the signature token can have a range of useful properties in commercial transactions.

REFERENCES

1. Lipis, A.H. 'Costs of the current U.S. Payments System', *Comm. ACM*, **22**, No. 12, 644–647, December 1979.
2. *Personal identification number management and security*, X9.8, ANSI X9 Secretariat, American Bankers Association, Washington D.C., 1982.
3. *PIN Manual; A Guide to the use of Personal Identification Numbers in Interchange*, Master Card International Inc. New York, September 1980.
4. Meyer, C.H. and Matyas, S.M. *Cryptography: a new dimension in computer data security*, John Wiley and Sons, New York, 1982.
5. Houghton, M.R. *An introduction to Electronic Fund Transfer Techniques*, Communications International, August 1980.
6. Meyer, C.H. and Matyas, S.M. 'Some cryptographic principles of authentication in Electronic Funds Transfer', *Proc. 7th ACM/IEEE Data Communications Symposium*, IEEE Computer Society Press, 1981.
7. Beker, H.J., Friend, J.M.K. and Halliden, P.W. 'Simplifying key management in electronic funds transfer point of sale systems', *Electronics Letters*, **19**, No. 2, 442–444, June 1983.
8. Australian Standard 2805 Electronic Funds Transfer — requirements for interfaces, Part 6.2.
9. Raubold, E. *Project proposal 'open shops for information services'*, (unpublished) GMD, Darmstadt, February 1982.
10. Needham, R.M. 'Adding capability access to conventional file servers', *SIGOPS Review* **13**, No. 1, 3–4.
11. *The Danish Securities Centre Act*, Act No. 179, of May 14th 1980.
12. *Legal aspects of automatic trade data interchange*, ISO/TC97/SC16/WG1, document N92, International Organization for Standardization, March 1983.
13. Brüer, J.-O. 'Original copy in the electronic world', *Information security in a computerized office* Part II Linkoping Studies in Science and Technology, Dissertations, No. 100, Linkoping University, Sweden, 1981.

Chapter 11 DATA SECURITY STANDARDS

11.1. INTRODUCTION

A characteristic of advanced technologies is the use of standards, which are needed for many reasons; safety in the case of lifting gear, durability in the case of drive belts and inter-changeability in the case of nuts and bolts. Because the market for the products of an advanced technology is worldwide, the process of standardization is international.

To clarify our terms it must be understood that the English word *standard* can mean two things which are distinguished in other languages. It can refer to standards of measurement such as the units of mass, time, radiation, illumination etc. The National Bureau of Standards (NBS) illustrates this meaning of the word. The other kind of standards which we are discussing in this chapter are known more specifically as *norms* and in many European languages 'normalization' is what we are calling 'standardization'. We shall, of course, continue to use the English word.

In the field of information technology the most frequent purpose of a standard is *interworking*. By this we mean that equipments designed by different teams and in different countries should be able to work together, either by direct connection using plugs and sockets or at a distance through communication networks. Interworking may seem very simple to the user, who would expect a visual display unit to plug into a computer and work. In fact, this level of interworking is very complex and has only partly been achieved. It requires a whole series of standards covering the physical properties of the plug and socket, the electrical signals which pass between them, the representation of binary digits, the procedure for exchanging bits, the representation of messages, their format, syntax and semantics, and higher-level procedures which enable one equipment to negotiate with another the type of service it is capable of providing or wishes to obtain. These complex requirements form a hierarchy and resemble the way that language is built up from a hierarchy of conventions, at the lowest level phonemes and words, then language, with its own very complex structure, and above this the protocol by which we communicate. The analogy between the interworking of computer systems and the complexity of language is very striking and it underlines the inevitably complexity of the standards needed in information technology. Standards of safety, quality and performance are also needed but the interworking requirement is the most characteristic feature of computers and communications.

Standards can completely change the nature of the market in computer components and equipment. Without standards, large manufacturers can tie their customers into a system which is vertically integrated. This benefits large manufacturers in the short term but it splits up the market into small units so that the specialist manufacturer (such as the maker of peripheral devices) cannot grow.

With the aid of standards, specialization among manufacturers can develop and a market which might otherwise seem small, for example the magnetic media, can become big enough to give economy of scale to many competing firms. The competition brings prices down. In this way both manufacturers and users have an interest in standards.

In communications, the need for standards became obvious as soon as the telephone network became fully international. At first the signalling systems of national networks did not work together without a struggle to achieve compatibility at each boundary, but now everything that needs to be decided for interworking is thrashed out by the International Telephone and Telegraph Consultative Committee (CCITT). Instead of trying to resolve interworking problems after they have happened, CCITT is moving towards prospective standardization — developing the international standards in advance of the start of national services. This has its dangers, but is probably better than leaving standards too late.

In computer technology, standards were not welcomed for at least two decades. The big manufacturers discouraged them because they saw their hold on the peripheral market being loosened. Researchers disparaged them because they felt their freedom to innovate would be restricted. There was some merit in these arguments while the technology was immature. Resistance to standards on both counts has now largely ceased.

There are many interests at work, consequently there are standards bodies at both national and international levels and some are working primarily for government users. In this chapter we shall first describe some of the actors in this drama — the standards bodies working in the field of data security — and then compare their different approaches to various data security standards.

To discuss all the detail in each standard or draft standard would merely be to rewrite the standard or reproduce it verbatim. The standards documents must speak for themselves and there are dangers in an attempt to interpret them. What we can do is to guide the reader and perhaps help him to read the definitive documents with better understanding.

The standards authorities

The first moves towards standards for data security were made by NBS (later renamed National Institute of Standards and Technology (NIST)), which is part of the US Department of Commerce. Under the Brooks Act (Public Law 89-306) the Secretary of Commerce is charged with improving the use of automatic data processing in the Federal Government. For this purpose, NIST provides *Federal Information Processing Standards* (FIPS) and technical guidelines, among other things. This work is done by the Institute for Computer Sciences and Technology of the NIST whose headquarters and laboratories are at Gaithersburg, Maryland. The address for enquires about FIPS is

The Director
Institute for Computer Sciences and Technology
National Institute of Standards and Technology
Washington, DC 20234
USA.

Because the main work on DES and related subjects was done by NBS, rather

than NIST, we retain the designation 'NBS' for this aspect of their work. The standards and technical guidelines are available from

US Department of Commerce
National Technical Information Service
5285 Port Royal Road
Springfield, Virginia 22161
USA.

These standards are maintained by the Institute for Computer Sciences and Technology in the sense that they must be reviewed from time to time to take account of improving technology. Normally they are applicable to all Federal departments and agencies but the FIPS publications relating to cryptography do not apply where classified data is being protected. A number of FIPS publications and some guidelines are described later in this chapter.

In the field of telecommunications, Federal standards are produced under the control of the *Federal Telecommunications Standards Committee* (FTSC). These standards are issued by the General Services Administration and are sold by

General Services Administration
Specification Unit (WFSIS)
Room 6039
7th and D Street SW
Washington, DC 20407
USA.

Both FIPS and the Federal Standards (and Draft Standards) produced by FTSC have had a major influence on work towards international standards. The international standards work has not slavishly followed the US Government examples but both the Data Encryption Standard (DES) and many features of other standards have been incorporated in international draft standards.

National Institute of Standards and Technology and FTSC represent government interests. The forum for US standards generally is the *American National Standards Institute* (ANSI) whose history goes back to an earlier body established in 1918. In common with other non-government standards bodies, the standards it produces are widely respected but not mandatory. The address is

American National Standards Institute Inc.
1430 Broadway,
New York, NY 10018
USA.

Like other standards organizations, ANSI operates through a large set of technical committees, subcommittees and working groups. A special characteristic of ANSI is the support of the secretariats of committees by other organizations representing manufacturers' and users' interests. For example the secretariat for ANSI work on the Data Encryption Standard has been

Computer and Business Equipment Manufacturers Association
(CBEMA)
Suite 500
311, First Street NW
Washington, DC 2000
USA.

For the development of ANSI standards relating to security in financial institutions, the secretariat has been provided by

American Bankers Association (ABA)
1120 Connecticut Avenue NW
Washington, DC 20036
USA.

Major countries have official bodies for the production of national standards, in the sense of norms. Examples in Europe are

The British Standards Institution (BSI)
2 Park Street
London W1A 2BS
UK

Association Francais de Normalisation (AFNOR)
Deutsches Institut für Normung (DIN)

The British Standards Institution was founded in 1901 and its work on data security is entrusted to a committee IST/20 under the general heading of Information Systems Technology. Where possible BSI tries to align its national standards with international standards and sometimes it exerts its effort towards the production of an international standard before beginning the production of a British one. The main technical contributions to international work on data security standards have come from the bodies mentioned above, particularly ANSI, ABA and BSI.

International standards in data processing are produced under the joint aegis of the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC). The technical committee responsible for data processing standards is Joint Technical Committee 1 (JTC1), for which the secretariat is

ISO/IEC/JTC1 Secretariat
American National Standards Institute Inc.
1430 Broadway
New York, NY 10018
USA

The work of ISO/IEC in data encryption and its application falls under technical committee JTC1. The work began under the committee predecessor of JTC1, ISO/TC97, which set up its working group 1 in 1980. This was replaced in 1984 by subcommittee 20 (SC20). Therefore the designation of the ISO/IEC committee currently working on data security standards is ISO/IEC/JTC1/SC20; the secretariat for this subcommittee is

ISO/IEC/JTC1/SC20 Secretariat
GMD/Z2.P
Schloss Birlinghoven
Postfach 1240
D-5205 Sankt Augustin 1
West Germany

There are other international bodies, such as the International Telephone and Telegraph Consultative Committee (CCITT), producing standards in various fields

of technology. Where the interests of data processing and data communication merge, there is good co-operation between ISO and CCITT.

CCITT is the official body at which telecommunication authorities and operating agencies produce the standards which enable their various systems to interwork. The Committee, together with CCIR, the corresponding body for radio, forms the International Telecommunications Union (UIT) which is an organ of the United Nations. In the most recent CCITT study period work has begun on various data security standards, for example a Directory Authentication Framework. CCITT is particularly concerned with open systems interconnection standards up to layer 4, whilst ISO/IEC committees deal with all layers, 1 to 7.

In addition to the official bodies so far described, there are special interest groups which can have a big influence on standardization by being represented in CCITT or ISO/IEC and presenting views which are known to be a consensus of a large group.

In CCITT there is an interest group which is the European Committee for Posts and Telecommunications (CEPT). Telecommunication authorities and operating agencies in Europe meet in CEPT to generate drafts which, when presented to CCITT, are understood to represent a consensus, in spite of the unofficial status of CEPT. In a similar way, the European Computer Manufacturers Association (ECMA) has had an influence on the work of ISO/IEC by the work it has done to produce standards.

The number of interests involved and their complex interaction makes it impossible to give a comprehensive account of standardization work, even in the relatively narrow field of data security. Data security standards form a hierarchy based on the application of encipherment; initially the emphasis was almost entirely on the Data Encryption Standard (DES), but the work is now broader based, specifying no particular encryption algorithm; public key algorithms, discussed in chapter 8, are entering the picture. The next higher layer of standards can use any block cipher, such as the DES, and this layer contains the 'Modes of Operation' which were described in chapter 4; definition of modes of operation may also be necessary for public key algorithms. Above the modes of operation are the procedures for employing encipherment (and authentication) in open systems interconnection; the ISO standard which describes the security architecture, ISO 7498/2, was discussed at length in chapter 6. ISO 7498/2 describes the security services which may be required in the OSI context and suggests where these may be located in the hierarchy; however it does not prescribe the detailed implementation of these services. Further standards will be needed for key management and possibly for other purposes not yet defined. Also there are specific applications of data security for which standards are produced, for example the authentication of financial messages, where the ABA and other banking organizations have special interests. In the following sections we shall describe some of the standards work which is taking place in these areas.

11.2. STANDARDIZATION RELATED TO THE DATA ENCRYPTION STANDARD

The early history and the nature of the DES are described in chapter 3. Subsequent work on DES in the national and international bodies has in no way departed from the algorithm that was first published in definitive form in FIPS Publication

46.[1] However, there are a few ways in which the algorithm is implemented which have subsequently been changed.

Publication 46 states 'the algorithm specified in this standard is to be implemented in computer or related data communication devices using hardware (not software) technology'. There is no such requirement in the American national standard entitled Data Encryption Algorithm[2] which is ANSI X3.92-1981.

There are dangers in implementing any cipher by software in a processor which is not adequately protected, particularly one which is shared by many tasks or many users. It is notoriously easy for systems programmers or maintenance engineers to read any part of a store and this could be the part containing the key being used for the cipher (or, equivalently, the 16 DES sub-keys). This was, no doubt, the concern which led to the hardware requirement in FIPS 46. The Federal Standard allows the use of a microprocessor having its program in read-only memory.

The DES has a 56-bit key contained in a 64-bit key variable and in FIPS 46 it is stated that the other eight bits are 'set to make the parity of each 8-bit byte of the key odd, i.e. there is an odd number of ones in each 8-bit byte'. In practice, many users of the DES have departed from this rule, allowing the extra bits to be arbitrary or using them for other purposes. The ANSI Data Encryption Algorithm states in section 3.5: 'one bit in each 8-bit byte of the KEY may be utilized for error detection in key generation, distribution and storge. Bits 8, 16, ... 64 are for use in ensuring that each byte is of odd parity.'

These sentences are exact copies of sentences in FIPS 46, page 16, and in this respect the two standards differ only in the emphasis given to the parity requirement in the FIPS explanation section and in the ANSI foreword, which are not parts of the standards.

Until 1987 an international standard corresponding to the DES was in preparation. Publication of this text has been abandoned by ISO for reasons which we shall discuss shortly. In this text no requirements concerning key parity conditions were specified. The text stated that: 'bits 8, 16, 24, 32, 40, 48, 56 and 64 of the key-block do not take part in the algorithm. Their use is not specified in this standard.' The text noted that: 'in some applications they have been used as odd parity bits'.

The name of the standard was modified a little during its adoption by ANSI. It is not usual to have the word 'standard' in the title of a standard because this is unnecessarily repetitive. Therefore the US national standard is called 'Data Encryption Algorithm' which can be shorted to DEA. The ISO name for the algorithm was to have been 'Data Encipherment Algorithm No. 1', shortened to DEA1; this title recognized the intention of publishing further encipherment standards.

In the event, as we have indicated already, ISO decided not to publish the DEA1 as an international standard. The main reason for this was a feeling that publication as an ISO standard would lead to the algorithm becoming even more popular and widely used. There is no doubt that the algorithm has been adopted as a component in many secure systems, not least by the international banking community. The fact of this widespread adoption has led to unease that the target presented by the large number of systems dependent on DES would be attractive to criminal cryptanalysts; the potential gain accruing from 'breaking' the algorithm could be immense and the possible damage to systems credibility could

be catastrophic. Any action further encouraging even more dependency on one algorithm could be undesirable. This belief led ISO to decide against publishing DEA1 and to replace the publication by a register of encipherment algorithms to be operated on behalf of ISO. Publication of the register is intended to encourage a degree of diversification in choosing encipherment algorithms for particular applications. Later in this chapter we shall examine the nature of the register as defined in the draft standard describing its structure and functionality.

Several useful texts have been published by the National Bureau of Standards regarding the DES. 'Guidelines for Implementing and Using the NBS Data Encryption Standard'[3] is a useful introduction to data encryption and it describes security threats, the implementation of the algorithm, key management and transparency. It also describes how to apply the DES with a limited character set — the method which was described on page 90.

'Validating the Correctness of Hardware Implementations of the NBS Data Encryption Standard'[4] describes a test bed for validating an implementation of the DES. The kind of tests suggested by NBS in this publication ensure that any 'normal' implementation conforms entirely, but for a cipher device we must also consider a malicious implementation which is designed, under very special conditions, to betray its key. Such an implementation with a 'Trojan Horse' could be designed to pass the NBS tests with very high probability. Therefore, for reasons of strict accuracy, it is not possible to give a definition of compliance testing. A device complies with the standard if it produces the correct output for all possible inputs and this means that the *mechanism* of the device must be inspected. Since the mechanism is not defined, only the result, there is no way to define the inspection process.

Federal Standard 1027 — General security requirements for equipment using the DES

Federal Standard 1027[5] is mandatory for all DES cryptographic components, equipments, systems and services used by US Government departments and agencies (the mandate was confirmed by a US Treasury Directive dated 16 August 1984). For other users and manufacturers it provides a useful guide to the security features which go beyond the cipher and its modes of operation. It covers physical security, locks, mounting, key entry, alarms, tests, controls and indicators among other things.

Key entry can be carried out *manually* and then the key is entered as 16 hexadecimal digits starting with bits 1 to 4 of the 64-bit key variable. The alternative is entry from a *key loader* and for this purpose a synchronous serial data interface is defined in the specification. The entry of keys is under the control of a lock which requires a physical key to be entered and turned. The same lock controls entry of the initializing variable (IV) when this is entered, along with the key, for cipher block chaining mode. It would seem logical to enter the IV on the same interface as the key variable but this possibility is not mentioned in the specification. Federal Standard 1027 requires odd parity in the octets of the key variable, in strict accordance with the DES of FIPs 46.

The standard FS 1027 specified the automatic testing of its internal functions

which the equipment must carry out. One alternative is to duplicate the encryption equipment and compare the results before output. A means must then be provided to test the comparator circuits by forcing an error. An alternative is to carry out a so-called 'S-box test' each time a new key is entered and a 'checkword test' each time an IV is used, i.e. at the beginning of each chain. Alternatively the S-box test can be carried out for each new IV usage. The *S-box text* is a test of DES operation with a set of values which guarantees that all the S-box entry combinations have been applied. Such a set is part of the NBS validation test.[4] The *checkword test* operates by enciphering a known 64-bit input word with the key and storing both the input word and the corresponding cipher text. For the checkword test, the same encipherment is carried out again. The data for this test have to be renewed every time a new key is installed. The comparator used in the checkword test must also be capable of checking by forcing an error. Whichever comparator is used (on all encipherments or on the checkword test only) this comparator itself should be checked no less frequently than each new key installation.

Careful testing is needed to keep an encryption system secure because the failure to encipher correctly could send out plaintext on the line until the intended receiver reported the problem.

A *bypass* mode in an encryption equipment means that encipherment does not take place and plaintext goes out on the channel that would normally carry ciphertext. At the receiving end, the same bypass mode allows the plaintext to reach its destination. Clearly this is a very dangerous procedure; it is used only for diagnostic tests or in dire emergencies where the message is more important than its secrecy. Bypass mode is an optional feature in FS 1027 but it *must* be under the control of a lock and must be announced by a status indicator.

The standard concludes with requirements for electro-magnetic interference and compatibility using criteria from US military standards. There are many other details concerning alarms, indications, key handling, stand-by mode etc.

Using Federal Standard 1027 as a starting point, British Telecom authors have produced a more comprehensive document entitled 'General Security Requirements for Cryptographic Equipment: a Code of Practice'; a second document addresses operational aspects. This material has been further developed and published by the British Institution of Radio and Electronics Engineers.[6] Publication of a British Standard for physical security is being considered by BSI.

A register of cryptographic algorithms

In an earlier section we observed the ISO decision not to proceed with the standardization of cryptographic algorithms. In place of publishing standard algorithms ISO proposes to establish a register of cryptographic algorithms. The procedures under which the register is expected to operate are being developed in a draft international standard[7]; these procedures will specify how the register will be used by those wishing to make entries in the register and by the Registration Authority. The register will serve as a common reference point for the identification of cryptographic algorithms by a unique name. It will also contain the basic parameters associated with a register entry. The main purpose of the

register is to enable entities (such as communicating entities in OSI) to identify and negotiate an agreed cryptographic algorithm.

The Registration Authority's role will be to add, modify and delete register entries in accordance with the rules provided, to assign ISO-entry names to registered cryptographic algorithms as needed and to update and to have published the register when required. Note that the Registration Authority does not evaluate or make any judgement of the quality of protection provided by any registered algorithm; thus no guarantee of security quality is implied in the presence of an algorithm on the register.

Submission of algorithms for registration will be via the national member body of ISO in each country (BSI, ANSI, DIN, AFNOR, etc). Likewise any alterations to entries or deletions of entries will be initiated via the ISO member bodies.

The register entry details will include:

(a) a formal ISO-entry name for the algorithm
(b) proprietary name, if any
(c) intended range of use
(d) cryptographic interface parameters (block size, key domain, etc.)
(e) set of functional test words
(f) identity of organization requesting registration
(g) dates of registrations and modifications
(h) whether the subject of a national standard
(i) patent licence restrictions
(j) export restrictions
(k) list of references to associated algorithms
(l) description of the algorithm
(m) statement of modes of operation
(n) other information

It should be noted that items (j) to (n) are optional; some or all may not be present.

At the time of writing nominations for Registration Authority had been called for by ISO and we understand that two member bodies have put forward candidate organizations. The present state of the draft international standard is that agreement on its text has not been reached and further drafting will now take place.

11.3. MODES OF OPERATION

We have described in chapter 4 the four modes of operation which have been the subject of several standards. The original publication was FIPS Publication 81 entitled *DES Modes of Operation.*[8] This was adopted, apparently without change, by ANSI as *Modes of Operation for the Data Encryption Algorithm.*[9]

The modes of operation standard prepared by ISO was published in August 1987.[10] Whereas the FIPS and ANSI documents relate specifically to DES, the ISO standard is worded to apply to any 64-bit block-cipher algorithm.

We need add nothing more to the description in chapter 4 of the electronic code book (ECB) mode of operation and cipher block chaining (CBC). For CBC the standard does not specify how to deal with the incomplete final block (if there

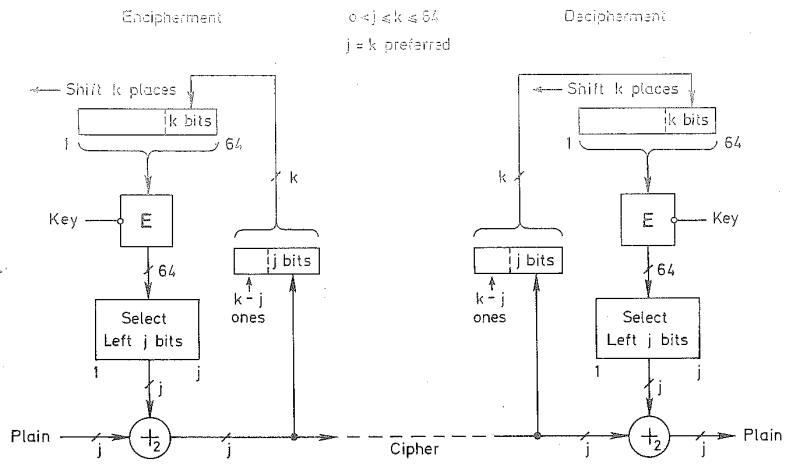

Fig. 11.1 Detail of the cipher feedback standard

is one) and leaves this to be treated in the individual application. For CBC an IV of 64 bits is used.

Cipher feedback (CFB) in these standards is a little more elaborate than the scheme described in chapter 4, which we illustrated only for 8-bit feedback. Figure 11.1 shows the CFB mode as it is described in the standard produced by ISO. In this figure, the feedback path is of width k bits. (The notation k must not be confused with the key.) In the shift registers a left shift by k bits takes place each time k bits are fed back. This creates the new input for the encipherment algorithm. From the output, the left j bits are selected and used for Vernam encipherment of the plaintext character. The difference between the character length j and the feedback length k is made up by $k-j$ ones which are attached to the left of the cipher character in the feedback path. It is recommended in the ISO standard that $k=j$ so that this difference between feedback width and character does not exist. The only case for which it is known to be used is a 7-bit character with an 8-bit feedback width.

For the IV the ANSI standard allows any number of bits to be chosen. This value is placed in the least significant or right-hand end of the shift register and padded on the left with zeros. Federal Standard 1027 requires an IV length of at least 48 bits in CFB operation.

In the ISO standard the 64 bits which fill a register to start any mode of operation are called the *starting variable*. The way these are derived from the given IV, which may be shorter, is not defined.

Output feedback (OFB) mode, sometimes known as 'output block feedback', is defined in FIPS 81 and the ANSI standard with a feedback path of k bits. We demonstrated in chapter 4 that OFB with feedback of less than 64 bits is insecure and that 64-bit feedback should always be used. The ISO standard allows only 64-bit feedback in OFB mode, but does provide that the leftmost j bits can be selected from the output block if this is convenient for the data stream being enciphered. Figure 11.2 shows, on the left, the OFB configuration in FIPS 81 and

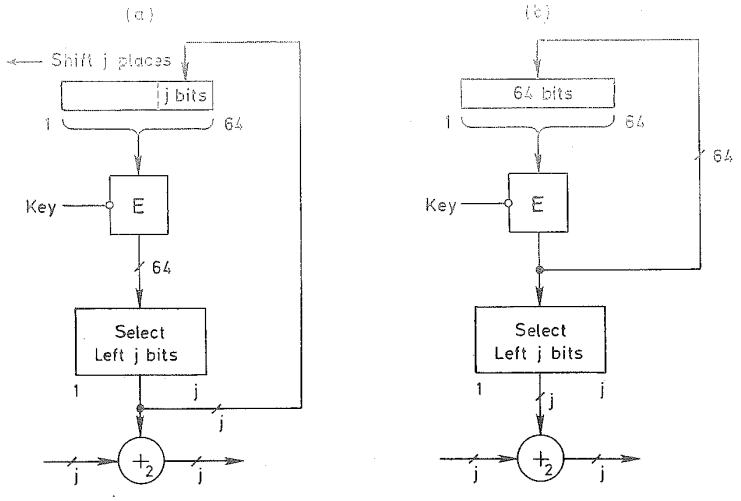

Fig. 11.2 Two versions of output feedback

the ANSI standard. The right side of the figure shows the configuration in the ISO standard. Since OFB mode is frequently used for high-speed synchronous operation it is unlikely that anything less than 64-bit output will be used in practice.

Note that ISO has now a draft standard[11] in preparation which provides definition of modes of operation for an n-bit block cipher; this carries the process of generalization one step further.

11.4. ENCIPHERMENT IN THE PHYSICAL LAYER OF DATA COMMUNICATIONS

One of the first international standards to be agreed in the area of secure communications was ISO 9160,[12] entitled 'Information Processing — Data encipherment — Physical layer interoperability requirements'. The physical layer is the lowest layer of OSI at which encipherment can be performed. Usually the encipherment can be continued only over one link of a network, but it has the advantage that all data bits are enciphered, making it a useful tool for traffic flow confidentiality.

The standard covers both synchronous and asynchronous or start/stop operation. In synchronous operation every element transmitted is enciphered. For start/stop operation, the start and stop elements which frame the characters transmitted must be preserved and only the data bits (normally 8 of them) are enciphered. As a consequence the presence of start/stop characters on the line is made obvious. The whole of the data including headers is enciphered but the quantity of characters carried remains obvious. To complete the traffic flow confidentiality in this case, dummy traffic would have to be added.

During the development of ISO 9160 the use of the DES algorithm in one-bit cipher feedback was assumed. In the eventual standard no algorithm is stated, because ISO's policy has changed. Even the mode of operation is left open but

the algorithm should be 'applicable on a single bit or single character at a time basis, as appropriate to the physical layer service provided'. In practice this will mean using one-bit cipher feedback or 8-bit cipher feedback.

For synchronous operation one-bit cipher feedback is ideal because all framing problems are solved. Its only problem might be the achievable data rate. For start/stop operation with 8-bit characters there are the alternatives of one-bit or 8-bit cipher feedback. Systems that have been built favour one-bit cipher feedback for its simplicity. They claim 'protocol transparency' since the only part of the protocol that may need adjustment is the setting up of the channel before transmission.

The use of the cipher feedback mode of operation as specified in ISO 8372 entails the prior transmission of an initializing variable or IV. Annex B of ISO 9160 describes the structure of the IV and its transmission for the use of DES (described here as DEA, i.e. ANSI X3.92). Briefly, for synchronous operation 48 bits are sent and then padded out with zeros on the left to form the 64-bit starting value or SV. As an alternative, all 64 bits can be sent. For start/stop operation the necessary number of characters are sent so that at least 48 bits of IV are obtained, padded out as before. There are two options giving, respectively, minimum IV lengths of 60 and 64 bits. Details are in Annex B which is an integral part of the standard.

The physical arrangement is, in principle, that shown in Figure 11.3, which shows a point-to-point connection with and without encipherment. The aim is to make the *data encipherment equipment* (DEE) as far as possible transparent; therefore it is placed between the data terminal equipment (DTE) and the data circuit terminating equipment (DCE) which is typically a modem. Consequently the DTE/DCE interface is 'split' and the DEE must present the appearance of a DCE on one side and a DTE on the other. In practice it is often more economical to incorporate encipherment in one of the other units, so that only a single DCE/DTE interface remains. To cover all versions, the standards describe encipherment as shown in Figure 11.3 and cover the other cases by requiring them to operate functionally in the same way.

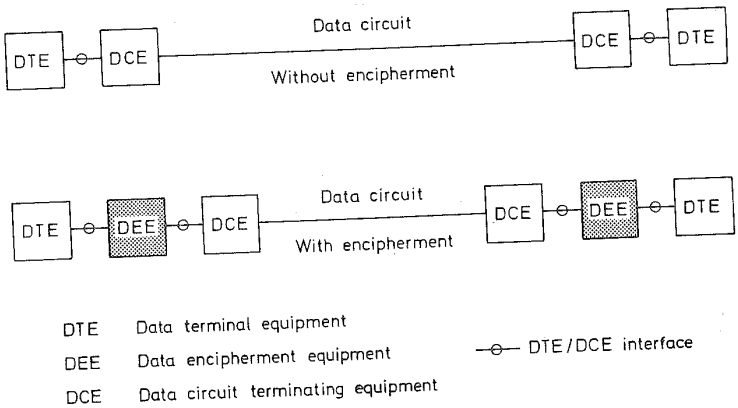

Fig. 11.3 Location of physical layer encipherment devices

Principles for encipherment at the physical layer

Encipherment is applied to the binary stream or stream of characters which is the physical-layer data service unit. Since the aim is to provide bit transparency, all versions of this standard agree in requiring 1-bit CFB. In the case of synchronous transmission every bit is enciphered in this way. In the case of start/stop transmission the start and stop elements must be preserved so that only the data-carrying bits in between them are enciphered. At the end of each character, the encipherment process stops, the stop bit is transmitted and nothing happens until the next start bit arrives; this is transmitted unchanged and encipherment begins again and continues until the next stop bit is reached. The notation employed in our figures is shown in Figure 11.4 with the usual convention that the *mark* condition corresponds to 1 and the *space* condition corresponds to 0. The start elements are represented by space and the stop elements by mark, which is also the condition of the line between characters. The shading indicates the enciphered data, which is framed by the start and stop elements.

Before encipherment and decipherment can begin, the IV must be transmitted over the line and this is sent in clear for CFB operation.

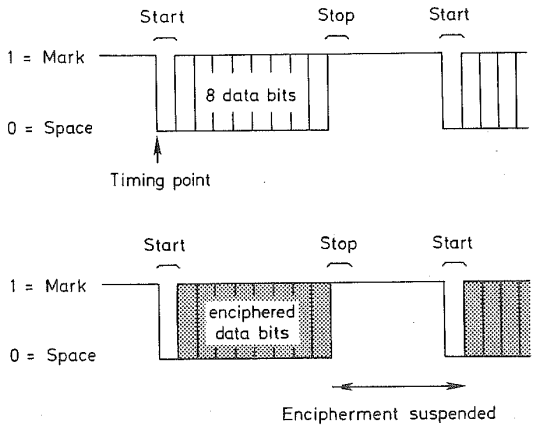

Fig. 11.4 Encipherment of start/stop characters

Signalling the start of transmission

Transmission can begin as soon as a connection has been made across the DTE/DCE interface by the exchange of suitable signals. We will use V24 terminology. New data networks employ CCITT recommendation X20 for asynchronous circuits and X21 for synchronous circuits and there are also recommendations X20 bis and X21 bis which allow the use of an interface similar to V24 in connection with these new networks.

The data encipherment equipment (DEE) is almost transparent to the procedure used to set up the circuit. Most of the signals involved are passed through from one interface to the other. The standard has been written to cover the older

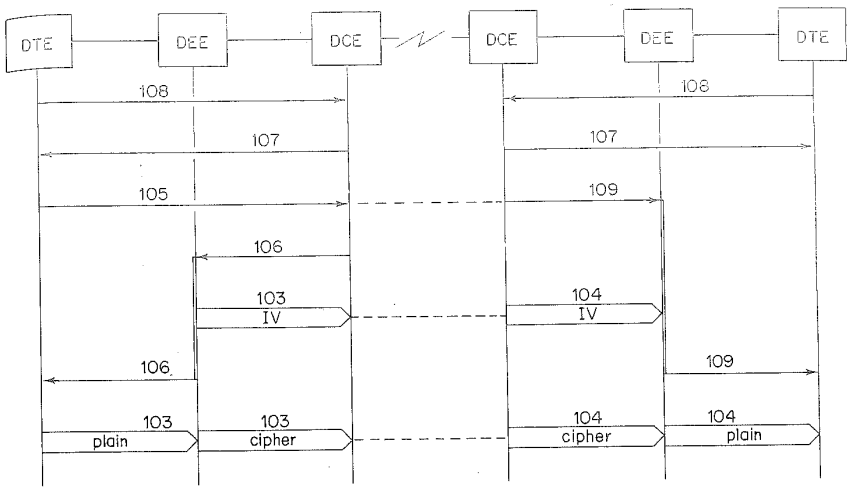

Fig. 11.5 V24 interchange circuits during circuit establishment

DCE/DTE interface V24, which allows both synchronous and start/stop working and the newer X20 (start/stop) and X21 (synchronous) interfaces. The transitional versions X20 bis and X21 bis can be treated as versions of V24.

The main new requirements are to let the DEE know when the circuit is ready and to ensure that the transmission of user data is held up until the IV has been sent and received. Figure 11.5 shows the signals employed in the V24 interface.

107 (data set ready) and 108 (data terminal ready) must be on and, if the DCE requires it, 105 (request to send). At the receiving end 109 (data channel received line signal detector) alerts the DEE but is not passed on yet to the DTE. The DCE at the sending end returns 106 (ready for sending) but this also is held up by the DEE. The DEE then sends the IV over line 103 (transmitted data) and this is received on line 104 (received data) and absorbed by the receiving DEE.

These preliminaries having been completed, the sending DEE forwards the 106 signal and the receiving DEE forwards the 109 signal, after which the two DTEs can communicate normally with the DEEs doing the encipherment and decipherment transparently. Figure 11.6 shows the way that the IV is followed by enciphered data in start/stop operation.

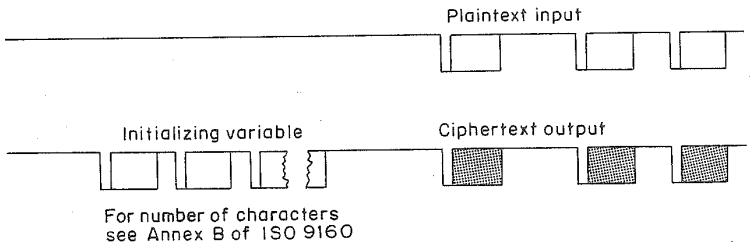

Fig. 11.6 Start of encipherment at the sending data encipherment equipment (start/stop)

For synchronous operation, the IV is sent as one continuous block of data, preceded by a single zero bit which marks its beginning, since the line is earlier in a MARK or '1' condition. Following the transmission of the IV, there are two alternatives. In alternative A the first enciphered data bit can follow immediately. In alternative B, the IV is followed by a '1' state on the line for 10 ms to 50 ms, then a single-zero bit heralds the start of transmission of enciphered data.

The physical connection is broken at the end of transmission by the 107 or the 108 circuits going to the off state. Potentially, the DEE can break the connection via both of the circuits simultaneously, but this is a fault condition.

There are corresponding rules, given in the standard, for the start and finish of transmission using the X20 or X21 interfaces.

We have now described the normal operation of the system. In addition, the standard covers how the break condition for start/stop operation is handled and the use of 'bypass control' so that loopback tests can be carried out.

Treatment of the break signal

In start/stop operation it is possible under certain fault conditions for the line to go into a *space* condition for longer than the duration allowed in any character. This can also be used as an 'out of band' signal though this is less used in present-day systems. There must be some convention on how the DEE treats the break signal and it should be, as far as possible, transparent to it. Two possibilities are provided as options in the ISO document, known as Classes A and B, which are illustrated in Figure 11.7.

In Class A operation, the beginning of the break signal is interpreted quite naturally as the start element followed by a character consisting entirely of zeros. Consequently the DEE output consists of enciphered zeros until the point where the input can detect the absence of a stop element. This is recognized by the DEE as signifying a break condition, encipherment is halted and the zero condition is transmitted on the DEE output. At the receiving end, the enciphered zeros are correctly deciphered, but the absence of a stop element signals that a break condition is present so the deciphered zeros can be followed immediately by a continued zero condition while the break signal lasts.

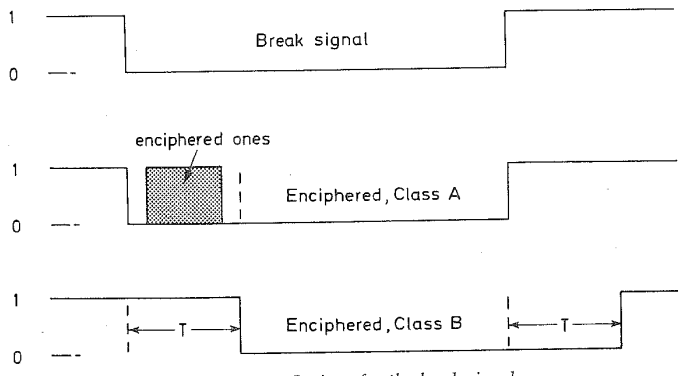

Fig. 11.7 Options for the break signal

In the alternative Class B, the DEE is transparent to the break signal which passes through it unchanged. To achieve this, the absence of the stop element must be detected before any character can be sent. This implies that a DEE operating in this manner must receive the entire character before a character is transmitted. That is, the DEE must have at least a one-character delay both in encipherment and decipherment, shown as T in the figure.

The option of bypass control

In the operation of modems on telephone circuits, the diagnosis of line faults is carried out by inducing a 'loop-back' in one of the modems. This can be a local loop back which tests nothing but the circuit from DTE to DCE and back again or a 'remote loop back' which takes place at the distant modem and tests the entire line without involving the equipment beyond the DCE at the far end. These provisions enable some types of fault to be distinguished without having to call an operator at the far end. The presence of data encipherment equipments would upset this diagnostic method.

To allow this diagnosis in the presence of a DEE when the V24 type of interface is used there is a *bypass* facility which suspends encipherment and transmits the plaintext through the DEE. This bypass is brought into operation by a 'test mode' circuit on the DTE/DCE interface, either 140 or 141 but not both. The occurrence of either signal is responsible for selecting the bypass mode. The appropriate loop-back signal is transmitted through the DEE to the DCE. In this way, the bypass mode should not occur without the loop back taking place. Figure 11.8 illustrates these two bypass–loop-back conditions.

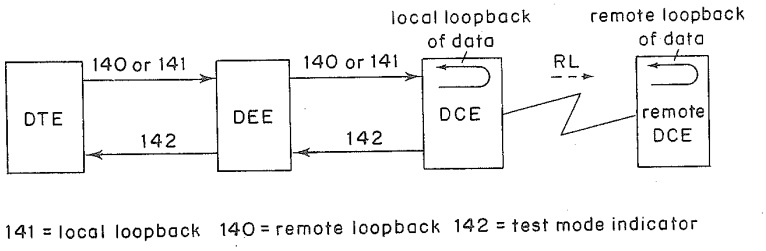

Fig. 11.8 Local and remote loopback configurations

Bypass control is an option which does not have to be provided on data encryption equipments to give compliance with the standard but the existence of this option in the standard will encourage manufacturers to include it. In order to be safe against misuse, bypass must be controlled by a lock with a physical key.

Characteristics of encipherment in the physical layer

The form of encipherment defined in ISO 9160 is well suited to fitting into a data network as an afterthought. It disturbs the existing system to a minimal extent. The delay due to encipherment and decipherment is about one bit time for each,

or one character time for each if 8-bit CFB is used with start/stop. Setting up a circuit is delayed by transmission of the IV, but this is significant only for half duplex operation with frequent turn-around. In the case of a leased line, no set-up delay will be apparent.

The extension of transmission errors due to the CFB mode of encipherment may be a problem on analogue circuits. It can be avoided by using an 'additive' stream cipher, but then the need for detecting loss of synchronization and recovering synchronization further complicates the system and prevents full transparency.

The standard does not ensure successful interworking unless the two ends make prior agreement on:

Cipher algorithm
Mode of operation
Size and format of IV
Break option (start/stop)
Immediate/delayed option (synchronous)
Use of bypass facility

Annex B of the standard, section B.4, describes how the choice of the various options should be codified.

ISO 9160 is the basis for a useful cipher facility that is relatively easy to install and should result in encryption products that have a wide range of use. Extension to encompass the ISDN interfaces is an obvious future task for ISO.

11.5. PEER ENTITY AUTHENTICATION

In secure communication two types of authentication are encountered — message authentication (message integrity) and peer entity authentication. The former has been discussed extensively in chapter 5. In the latter the aim is to allow communicating entities (user application processes for example) to obtain mutual assurance that both parties are genuine. Various methods of peer entity authentication have been proposed, but a basic feature of all systems is that the entities satisfy each other that they possess some secret that can be verified. Secret key cryptosystems and public key cryptosystems can be applied to achieve this type of authentication, as can also the recently conceived zero knowledge protocols.

ISO is currently developing several standard texts for peer entity authentication. One of these[13] makes use of an n-bit secret key algorithm. Within this draft standard four distinct mechanisms are identified. The first assumes that genuine communicating entities are in possession of the same secret key before authentication is attempted; the protocol consists of challenges that establish to the mutual satisfaction of both parties that each is in possession of the same secret key. The second does not start from the same premise; at the start no common secret key exists, but a third party, an authentication server, is invoked to make up the deficiency. The third protocol improves on the second by assuming that all parties have reliable clocks available; this simplifies the task of preventing fraud by replay of messages. The fourth protocol deliberately sets out to minimize the security relevant information that flows between the authenticating entities.

To illustrate the principle of peer entity authentication we shall explain the first of the above mechanisms. The initiating entity A generates a random number,

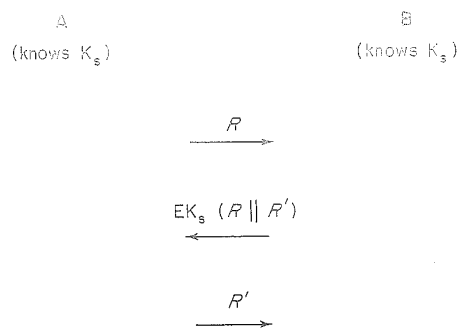

Fig. 11.9 Peer entity authentication with secret key algorithm

R, see Figure 11.9, and sends this to entity B in clear. Entity B prepares a reply in which A's random number R is concatenated with a new random number, R', generated by B. Before sending the reply B enciphers it with key *Ks*, previously agreed by some unspecified method between A and B. When A receives the enciphered reply this can be deciphered with *Ks*, revealing the original random number R and the new one R'. To complete the authentication process A sends back R' in cleartext to B. The result of this process is that entities A and B are each assured that the other entity with which they are communicating is in possession of *Ks*. It is on this knowledge that the authentication is founded. Clearly the process of establishing *Ks* at the two entities in the first place must be secure, with no possibility of a hostile third party learning *Ks*. The remaining three mechanisms are more complicated but are based on the same principle.

Peer entity authentication using public key algorithms is somewhat more elegant. Two draft texts are currently being worked upon. The first[14] is concerned with a two-way handshake; certified copies of entity public keys are provided by an authentication server, the certification being based upon digitial signature. If we call the initiating entity A and the other entity B, the protocol proceeds by A sending a signed 'token' (which includes a time and date stamp, but no random number) to B together with the certified value of A's public key. Using the latter B can check the validity of the former. If satisfied, B responds with B's signed token which A can verify using the certified value of B's public key obtained from the authentication server. Only one message flows in each direction between the parties, hence the name 'two-way handshake'.

The second draft text[15] using public key algorithms makes use of a 'three-way handshake'. The three-way handshake protocol is distinguished from the two-way handshake protocol by the presence in each message of a random number challenge; each entity generates its independent random number. The authentication is based on signature of tokens containing the random numbers and three messages are necessary to complete the process. There is a close logical parallel between this public key method and the secret key method explained above.

Because of the close connection between the three peer entity authentication texts discussed above, it is proposed in ISO that the three texts shall be brought together into a unified text before final publication as an international standard.

In this connection it is noteworthy that CCITT has published a text[16] describing

a directory authentication framework. This text is extremely detailed and covers much the same ground as the three ISO texts; unfortunately it is not compatible with the ISO texts. However, it also goes far beyond these in discussion of issues such as key management and authentication servers; these issues are treated in ISO as separate work items and will be the subject of separate standard texts.

11.6. STANDARDS FOR DATA SECURITY IN BANKING

The work on standards development described in the foregoing sections is carried out at ISO within subcommittees of Technical Committee JTC1, notably SCs 6, 20 and 21. There is another area of substantial activity within ISO that is also producing standards for data security. This is within the domain of Technical Committee 68 on Banking Procedures; the subcommittees particularly charged with data security responsibilities are 2 and 6. The work under TC68 has been almost totally independent of that under JTC1.

Before giving a summary of the data security work within TC68 it is well to consider the work of the banking interests within ANSI, because ANSI standards have formed the basis of some of the standards in the data security area published by TC68. The ANSI committee working in the secure banking area is X.9 (which operates in close collaboration with the American Bankers Association). The following is a list of relevant standards produced within X.9:

X9.8 Personal Identification Number (PIN) Management and Security,
X9.9 Financial Institution Message Authentication (Wholesale),
X9.17 Financial Institution Key Management (Wholesale),
X9.19 Financial Institution Message Authentication (Retail),
X9.23 Financial Institution Encryption of Wholesale Financial Messages,
X9.24 Financial Institution Key Management (Retail).

Thus far TC68 has produced several standards in the data security area. The first of these was ISO 8730[17] which describes the logical procedure for message authentication; it is closely based on ANSI X9.9, but does not specify encipherment algorithms for use in the authentication process. The second TC68 standard was ISO 8731[18] which has two parts, each specifying a suitable algorithm for message authentication. The first part of ISO 8731 specifies the DES algorithm, using the ANSI standard X3.92 as a reference. The second part of ISO 8731 specifies the Message Authentication Algorithm which is discussed in chapter 5 of this book. TC68 has also published ISO 8732[19] on wholesale key management; this is based fairly closely on ANSI standard X9.17 and was described in chapter 6. Further standards are in process of development.

11.7. POSTSCRIPT

We close this account of data security standards with a postscript rather than with a set of conclusions because the work is ongoing in a very active sense. Indeed, as we write, the organization of data security standards work within ISO/IEC/JTC1 is undergoing a drastic reorganization. The same subjects will be worked upon that we have described here, with future extensions, but the distribution

of responsibilities between the various committees will be different. It is expected that a committee replacing SC20 will take responsibility for 'security techniques'. Committees with responsibility for particular communication protocols will look after security aspects of those protocols.

REFERENCES

1. *Data Encryption Standard,* Federal Information Processing Standards Publication 46, National Bureau of Standards, US Department of Commerce, January 1977.
2. *Data Encryption Algorithm,* X3.92, American National Standards Institute, 1981.
3. *Guidelines for Implementing and Using the Data Encryption Standard,* Federal Information Processing Standards Publication 74, National Bureau of Standards, US Department of Commerce, April 1981.
4. Gait, J. *Validating the Correctness of Hardware Implementations of the NBS Data Encryption Standard,* Special Publication 500/20, National Bureau of Standards, US Department of Commerce, November 1977.
5. *Telecommunications: General Security Requirements for Equipment Using the Data Encryption Standard,* Federal Standard 1027, Federal Telecommunications Standards Committee, April 1981.
6. Serpell, S.C. *Cryptographic Equipment Security: A Code of Practice,* Institution of Electronic and Radio Engineers, 1985.
7. *Data Cryptographic Techniques — Procedures for the Registration of Cryptographic Algorithms,* Draft International Standard 9979, International Standards Organization, 1988.
8. *DES Modes of Operation,* Federal Information Processing Standards Publication 81, National Bureau of Standards, US Department of Commerce, December 1980.
9. *Modes of Operation for the Data Encryption Algorithm,* X3.106-1983, American National Standards Institute, 1983.
10. *Information Processing — Modes of Operation for a 64-bit Block Cipher Algorithm,* International Standard 8372, International Standards Organization, 1987.
11. *Information Processing — Modes of Operation for an n-bit Block Cipher Algorithm,* Draft Proposal 10116, International Standards Organization, 1988.
12. *Information Processing — Data Encipherment — Physical Layer Interoperability Requirements,* International Standard 9160, International Standards Organization, 1988.
13. *Peer Entity Authentication Mechanisms Using an n-bit Secret Key Algorithm,* Draft Proposal 9798, International Standards Organization, 1988.
14. *Peer Entity Authentication Mechanism Using a Public Key Algorithm with a Two-way Handshake,* Draft Proposal 9799, International Standards Organization, 1988.
15. *Peer Entity Authentication Mechanism Using a Public Key Algorithm with a Three-way Handshake,* Draft Proposal 10117, International Standards Organization, 1988.
16. *The Directory — Authentication Framework,* CCITT Draft Recommendation X.509 (also DIS 9594-8), 1988.
17. *Banking — Requirements for Message Authentication (Wholesale),* Draft International Standard 8730, International Standards Organization, 1988.
18. *Banking — Approved Algorithms for Message Authentication, Part I, DEA,* International Standard 8731/2, 1988, International Standards Organization. *Part 2, Message Authentication Algorithm (MAA),* International Standard 8731/2, 1988, International Standards Organization.
19. *Banking — Key Management (Wholesale),* Draft International Standard 8732, International Standards Organization, 1988.

GLOSSARY

Active card
Thin plastic card of credit-card size containing a store, processor or both, and capable of interaction with a terminal, for example in a point-of-sale transaction. Also called a 'smart card'.

Active line-tap (or wire-tap)
Deliberate interference with a communication system to alter the signals, data or messages it carries. This includes introducing new data or messages, repeating those that were sent earlier, delaying or deleting messages. *See also* passive line-tap.

Algorithm
A precise statement of a method of calculation. It can sometimes be expressed conveniently in a programming language.

Alphabet
The range of values which may be taken by a particular unit of communication, such as a character. When a cipher employs one or more permutations of a character code, each of these is known as an alphabet.

ANSI
American National Standards Institute.

Arbitration
Settling a dispute by referring it to a third party. This arbitrator may take part in a procedure which enables disputes to be settled, as in the production and checking of an 'arbitrated signature'.

ASCII
American Standard Code for Information Interchange. Its full title is USASCII. This code is a national version of the ISO 7-bit coded character set and is mainly compatible with CCITT alphabet No. 5.

ATM
See automatic teller machine.

Authentication
1. Ensuring that a message is genuine, has arrived exactly as it was sent and came from the stated source.

2. Verifying the identity of an individual, such as a person at a remote terminal or the sender of a message.

See also: Data origin authentication and peer-entity authentication.

Authenticator
A number which is sent with a message to enable the receiver to detect alterations made to the message since it left the sender. The number is a function of the whole message and a secret key.

Automatic teller machine (ATM)
A machine at which customers can do business with their bank. Withdrawal of cash is a main part of the service, but the ATM is typically more than a 'cash dispenser' and allows, for example, presenting cheques for payment into an account and enquiry for the account's balance.

Bijection
A mapping from one set of entities $\{x\}$ onto another $\{y\}$ having the same number of elements, such that each x maps onto a unique y and each y is the mapping of a unique x, i.e. it is a one-to-one mapping.

Birthday paradox
How many people must there be in a group to make it likely that at least two of them have the same birthdate? The answer, twenty-three, appears paradoxical. This type of problem is significant for the understanding of security in some systems (*see* page 279).

Block
A block of data is an array or sequence of bits that are usually stored or transmitted together. Many such arrays have specialized names such as 'character', 'field' or 'packet' which suggest their function. Block is a neutral term and often signifies a fixed-length array of bits.

Block cipher
A cipher method which acts on a fixed-length block of data to produce a result which depends only on that block.

BSI
British Standards Institution.

By-pass mode
A mode of operation of data encipherment equipment in which plaintext passes through unchanged. Federal Telecommunications Standard 1027 requires by-pass mode to be controlled with a lock.

CCITT
International Telephone and Telegraph Consultative Committee. The letters come from the title in French. With the CCIR (radio) it forms the International Telecommunications Union (UIT).

Chain
A sequence of message units which are treated together for encipherment, decipherment or authentication. The units may be characters or blocks, for example. Chaining prevents the analysis of ciphertexts being applied to individual message units, or such enciphered units being reassembled in a different sequence to make a meaningful message.

Check field
A field added to a data item which provides redundancy in such a way that the most likely errors will be detected. Check fields are often used to catch keyboarding errors.

CID
Chain Identifier; a sequence number used in the plaintext of a chain which is to be enciphered. Its purpose is to guard against replay and deletion.

Cipher
A method of cryptography by applying an algorithm to the letters or digits of the plaintext. A distinction is made between a cipher and a code, the latter using a table instead of an algorithm. This distinction is not always clearcut.

Ciphertext
The enciphered form of messages or data.

Code
A method of cryptography which employs an arbitrary table (a codebook) to translate from the message to its coded form. Contrasted with a 'cipher'. The word also has non-cryptographic meanings as in 'character code'.

Complexity
The degree of difficulty of a calculation, measured by the number of elementary operations it requires. The nature of the elementary operations should be specified. Complexity is a function of the 'size' of the problem, which is measured by a parameter which should also be stated.

Computationally secure
Secure under the assumption that the computation needed to undermine the security is too extensive to be feasible.

Confidentiality
The property that information is not made available or disclosed to unauthorized individuals, entities or processes.

Cryptanalysis
The study and development of methods by which, without a prior knowledge of the key, plaintext may be deduced from ciphertext, given sufficient working material and computational power. One aim of cryptanalysis would be to discover the key, but that may not always be necessary.

Cryptography
The technique of concealing the content of a message either by a code or a cipher.

Cryptology
The science which includes all aspects of cryptography and cryptanalysis.

Cyclic redundancy check (CRC)
A kind of check field used to guard against errors. This is used in some CCITT recommendations and ISO standards.

DAC
Data Authentication Code. A name used for an authenticator in the proposed Federal Information Processing Standard on Computer Data Integrity. The word 'code' is misused in this term.

Data link layer
The layer of Open Systems Interconnection immediately above the physical layer. At this layer of procedure, error and flow control are introduced.

Data origin authentication
Corroboration that the source of data received is as claimed.

DCE
Data circuit terminating equipment. Data communication local lines typically end at an equipment on the subscriber's premises which belongs to the network and provides the 'customer interface'. This outermost part of the network is the DCE.

DEE
Data encipherment (or encryption) equipment.

DID
Data Identifier; synonymous with 'chain identifier'. A term used in the proposed Federal Information Processing Standard on Computer Data Integrity.

Digital signature
A number depending on all the bits of a message and also on a secret key. Its correctness can be verified by using a public key (unlike an authenticator which needs a secret key for its verification).

DTE
Data terminal equipment. The equipment which is connected to a data communication network in order to communicate. The DTE may be anything that communicates, for example a computer system.

EFT
Electronic Funds Transfer. A term used for various payment methods that employ communication or electronic storage. The most usual context is in point-of-sal- transactions.

Electronic code book
The straightforward use of a block cipher to encipher blocks of data. It likens the process to a code book in which the cipher is listed for all possible values of plaintext, but in practice the size of the 'code book' makes this infeasible. It should be used with caution.

Enemy
In this book it is a convention to refer to any adversary of the system as an 'enemy'. In security analysis the possibility of joint action by several enemies must also be considered.

Entity
A person or process taking part in a communications procedure. In 'Open Systems Interconnection' there can be communicating entities at any of the seven levels.

Error extension
See garble extension.

Exhaustive search
Solving a problem (usually that of finding the key that is in use) by searching through all possibilities, assuming that each can rapidly be tested and the true result (key) will show up.

Exponential function
Normally used to mean $\exp(x)$ which is e^x, but here we use it also for a^x as a function of x with a as a parameter, called the base.

FIPS
Federal Information Processing Standard. These are issued by the National Bureau of Standards and, though recognized more widely, are intended primarily for US Federal Departments and Agencies.

Flag
1. A part of the format of an item of data which contains a bit (or a few bits) which pertain to the use of the item as a whole. The item is said to be 'flagged'.
2. An easily recognized pattern of bits to delineate an item of data, such as a 'frame' in HDLC.

Garble extension
Noise in communication systems may produce errors in the ciphertext. When this is deciphered, the errors in the plaintext sometimes extend further than those in the ciphertext. This is garble (or error) extension.

Greatest common divisor (gcd)
For two numbers (positive integers), their greatest common divisor is the largest number which divides into both numbers exactly. Example: gcd (75,45) = 15.

Hamiltonian cycle
In a graph it is a cycle which includes all the points of the graph.

Hamming distance
Given fixed-length blocks of binary data, the Hamming distance between two such blocks is the number of bits in which their values are different.

```
Example:  1 0 1 1 0 1 0 1 1 1
          0 1 1 1 0 0 0 1 1 0
          * *     *       *   Hamming distance = 4
```

Hash function
A well-defined function of a message or block of data which appears to generate a random number. Care is needed in selecting hash functions because they are used for several different purposes.

Index of coincidence
Two texts can be placed together and corresponding characters compared for equality (coincidence). The proportion of coincidences among the characters is a very useful statistic. Friedman defined the index of coincidence as 'coincidences minus non-coincidences divided by total number of letters compared'.

Initializing value (or variable), IV
For the encipherment or decipherment of a chain it may be necessary, or desirable, to use a value (e.g. a block of bits or set of characters) which starts off the process. This is the initializing value or IV and is known to both sender and receiver.

Integrity (of data)
The property that data have not been altered or destroyed in an unauthorized manner.

Intelligent token
A small device incorporating a processor and store which can be carried by someone for identification and to make transactions with special terminals. One form of this token is the active card or 'smart card'.

ISO
International Organization for Standardization. The international forum at which representatives from national standards bodies (such as ANSI) meet to agree standards for world-wide use.

Key block
In the context of the Data Encryption Standard it is the 64-bit block of data which contains the 56-bit key. It is also known as the 'key variable'.

Key space
The range of all the values which a cryptographic key may take. A large key space is important for security.

Key variable
See key block.

Knapsack problem
A problem which is important in complexity theory, see page 235. A number of public key ciphers and some signature methods have been based on varieties of the knapsack problem.

Least common multiple (lcm)
For two numbers (positive integers) their least common multiple is the smallest number into which they both divide exactly. Example: lcm (75,45) = 225.

Line level encipherment
On-line encipherment of the data or messages passing over a single line of a communication system. As they leave the line, the data or messages are deciphered. This term does not correspond with OSI terminology, but typically involves either the physical or data link layer.

Longitudinal parity check
A parity check applied throughout a string of data, for example the modulo 2 sum of each of the 8 bits in a string of octets, which together form a 'check octet'.

MAC
Message Authentication Code. A name used for an authenticator, for example in ANSI Standard X9.9 'Financial Institution Message Authentication'. The word 'code' is misused here.

MDC
Manipulation Detection Code or Modification Detection Code. A redundancy check field included in the plaintext of a chain before encipherment, so that changes to the ciphertext (an active attack) will be detected. The word 'code' is misused here.

Modulo, modulus
The expression 'modulo m' is a statement that arithmetic with m as modulus is being used, see page 245.

Multi-drop
A communication line which connects to more than two stations so that (in principle) each may communicate with several others, according to an agreed procedure.

Negative file
In the context of payment systems, a file listing those accounts which are not to be allowed transactions because of some problem, such as a lost or stolen card or a denial of further credit.

Node
An equipment in a communication network at which two or more communication lines converge. Usually it refers to a switching node but, strictly speaking, a multiplexer or concentrator could be called a node.

One-time tape or pad
A tape containing random numbers or bits. Only the original and one copy are made so that one station can use it to encipher and another to decipher messages under the strict rule that each portion of the tape may be used only once. Another version is the one-time pad, where the numbers are written down on two identical pads; used pages are destroyed.

One-way function
A function $y=f(x)$ which is relatively easy to compute but much more difficult to compute the inverse. That is, given x it is easy to find y, but given a value y it is (except in a few cases) difficult to find *any* solution x of $y=f(x)$. The exception allows for the fact that, with luck, the x,y pair may already be known.

Open Systems Interconnection (OSI)
A system of standards which should make useful communication possible between any systems attached to a communications network.

Padding
When a message (or a record of arbitrary length) is divided into fixed-length blocks for any purpose, the last block may not be completely filled. The information used to fill the remaining part of the last block is padding.

Passive line-tap (or wire-tap)
Unauthorized reading of signals, data or messages from a communication system, without changing the signals. *See also* active line-tap.

Peer-entity authentication
Corroboration that a peer entity in an association is the one claimed.

Permutation
Changing the order (or sequence) of a set of data elements. The elements can be bits, characters, bytes, etc. Permutation can be used as one operation in a process of encipherment. *See also* transposition cipher.

Physical layer
The lowest layer of Open Systems Interconnection, in which physical phenomena (voltage, current, light waves) are interpreted as a sequence of bits or other units (such as start/stop characters).

PIC
Personal Identification Code. A sequence of symbols (letters and numbers) used to verify the identity of the holder of a bank card. It is different from a PIN in allowing letters to be used. The word 'code' is misused here.

PIN
Personal Identification Number. A sequence of decimal digits (usually four, five or six) used to verify the identity of the holder of a bank card. A kind of 'password'.

Plaintext
Messages or data which are not in an enciphered state and are therefore in their normal, readable form.

Power function
The function a^n is the nth power of a. This is considered as a function of a with n as parameter. Thus $y=a^4$ is the fourth power of a.

Prime
A prime number is a positive integer n which has no divisors except the trivial ones 1 and n. Sometimes 1 itself is included among the prime numbers, sometimes not. Non-primes are called 'composite numbers'.

Product cipher
A cipher employing both substitution and transposition, in some cases repeated alternately.

Pseudo-random
A sequence of data which are generated by an algorithm but nevertheless appear random and will pass the customary tests for randomness.

Public key
A cryptographic key used for encipherment but not usable for decipherment. It is therefore possible to make this key public.

Relatively prime
Two numbers (positive integers) are relatively prime if they have no common divisor except the trivial divisor 1. That is to say: their greatest common divisor is 1. Example: 65,24.

Replay
An attack on a cipher system (or authenticator) in which a message is stored and re-used later, replacing or repeating the original. There is a similar attack on stored data by restoring an item to an earlier state it once had.

Repudiation
Denial by one of the entities involved in a communication of having participated in all or part of the communication or denial of the content of the communication.

Residue, modulo m
The residue of x, modulo m, is the (positive) remainder r which is left when x is divided by m. It follows that $0 \leq r < m$. Finite arithmetic is the arithmetic of residues.

Round
The Data Encryption Standard algorithm consists of an initial permutation, then sixteen 'rounds', a register interchange, then a final permutation which is the inverse of the initial one. The sixteen rounds are identical operations but each uses its own sub-key *q.v.*

RSA
Rivest, Shamir and Adleman gave their names to a public key cipher which is usually known as the RSA cipher. There is a corresponding 'RSA digital signature'.

S-box
A substitution operation used in the Data Encryption Standard. This algorithm employs eight different 'S-boxes'.

Security policy
The set of rules for the provision of security services.

Seed value
A value (e.g. block of bits or set of characters) which is used to start a pseudo-random sequence generator and provide unpredictability. A seed must be randomly chosen and usually must be kept secret.

Session key
A cryptographic key which is used only for a limited time, such as one communication session, then discarded. Typically it is transported through the network under encipherment with another key.

Settlement
When banks send messages to effect customer payments, customer accounts are updated at once, but these changes do not constitute net payments between the banks because they are balanced in an account held by one bank for the other. The eventual payment between banks is called settlement and usually employs accounts at some other bank.

Smart card
See active card, intelligent token.

Starting variable
The ISO standard for 'modes of operation' distinguishes the initializing variable or IV from the 64-bit block which is loaded into a register before encipherment/decipherment begins. The latter is called the 'starting variable' and is derived from the IV.

Steganography
Concealment that a means of communication exists. Examples are the use of invisible ink or microdots.

Stream cipher
A cipher which is devised in such a way that an indefinitely long stream of data can be dealt with. It contrasts with a 'block cipher', but both kinds of cipher can be adapted to the other purpose.

Sub-key
In the Data Encryption Standard algorithm, the 56-bit key is used in a 'key

generator' to derive (by simple means) sixteen sub-keys, each of 48 bits, which are used in the sixteen 'rounds' of the algorithm.

Substitution, substitution cipher
Replacement of one unit of data (such as an octet or character) by a related one. A cipher in which each character is replaced by one derived from it is a substitution cipher. The substitution boxes (S-boxes) of the Data Encryption Standard each produce a 4-bit function of a 6-bit input.

S.W.I.F.T.
Society for Worldwide Interbank Financial Telecommunication, which provides a specialized communication network for banks, *see* page 289.

Synchronization
In the context of cryptography, synchronization means keeping in step the processes involved in encipherment and decipherment.

Test key
Used in banking to mean the same as authenticator. The secret key used in calculating the authenticator produces a confusion of terms, so the word 'authenticator' is preferable.

Time-memory trade-off
Calculations which require a large number of computational steps can sometimes be carried out with less computation at the cost of having to store a large number of intermediate results. This is trading time of computation for size of memory, called a time-memory trade-off.

Time stamp, time and date stamp
A value inserted in a message to give the time at which the message was originated. If necessary it is extended to include the date, becoming a time and date stamp.

Token
Something possessed by a person to assist in identifying him, etc. Its form could be a card, pen, key, etc. and it can have functions additional to identification, for example, storage, access control, payment.

Traffic analysis
When communication traffic is in cipher form and cannot be understood it may still be possible to get useful information by detecting who is sending messages to whom and in what quantity. This is traffic analysis.

Traffic flow confidentiality
Concealment of the quantity of users' messages or data in a communication system and their sources or destinations, to prevent traffic analysis.

Transparency
The ability of a communication system to pass messages through without any change. In data communication the requirement is to preserve a sequence of bits or characters.

Transposition cipher
A cipher which permutes a set of bits or characters from the plaintext. Mainly of historical interest as a cipher, but it can be used as a component of a more complex cipher.

Trapdoor
A feature in a cipher or a puzzle which, given the necessary information, enables it to be solved easily. The trapdoor is concealed from those who do not have the information.

Type I error
An error in the operation of an identity verification system which causes a genuine user to be rejected — a false alarm.

Type II error
An error in the operation of an identity verification system which allows a false user (someone claiming the wrong identity) to be recognized as valid — an imposter pass.

Unconditionally secure
Secure even though an enemy is assumed to have unlimited computing power available.

Unicity distance
The amount of ciphertext beyond which some knowledge of the statistical properties of the plaintext (its redundancy) allows the key to be deduced, given unlimited computing power. With less than this amount, there are multiple key values that are consistent with the given ciphertext.

Vernam cipher
A cipher which uses a sequence of bits (or characters) called the 'key stream' and enciphers by adding this stream modulo 2 to the plaintext. It deciphers in the same way with an identical stream. In the original Vernam cipher the key stream came from a loop of punched tape.

Weak key
A value of a cipher key which gives that cipher some special properties that might, in certain circumstances, weaken its security. The supposed weakness has to be related to a specific method of application of the cipher. There can also be weak key values for an authenticator or a digital signature.

INDEX

ecurity for omputer Networks

Second Edition

An Introduction to Data Security
in Teleprocessing and
Electronic Funds Transfer

D. W. Davies *Data Security Consultant*
and W. L. Price *National Physical Laboratory, Teddington, UK*

What the reviewers said about the first edition:

'This book is informative and a delight to read, and should find a position on the bookshelf of most computer users'
G. Moore, Electronics and Power

'I recommend it to all who are involved in the field . . . It should take its place on your shelf as one of the better reference books on network security'
Charles Cresson Wood, Computers and Security

'. . . an excellent book'
The Computer Law and Security Report

'An excellent and clearly written book'
Cryptologia

Attempts at blackmail, fraud and the theft of commercial secrets continue to pose threats to computer security. The rise in the use of information technology in business has considerably increased the opportunity for such crimes: hence the need for data security techniques is greater than ever.

In the second edition of this authoritative text, the authors have addressed the needs of modern users and brought the book right up to date. They have added material on open systems interconnection; key management (including the international standard in wholesale banking, likely to be the model for future standards); and electronic funds transfer at the point of sale (EFT-POS), with examples from some schemes with well-developed security (i.e. transaction key, derived key per transaction and the use of public key cryptography).

Of course, the book still covers excellently the basic principles of security, including cryptography and its applications.

ISBN 0-471-92137-8

9 780471 921370

JOHN WILEY & SONS
Chichester · New York · Brisbane · Toronto · Singapore

Davie
and
Price

WILEY