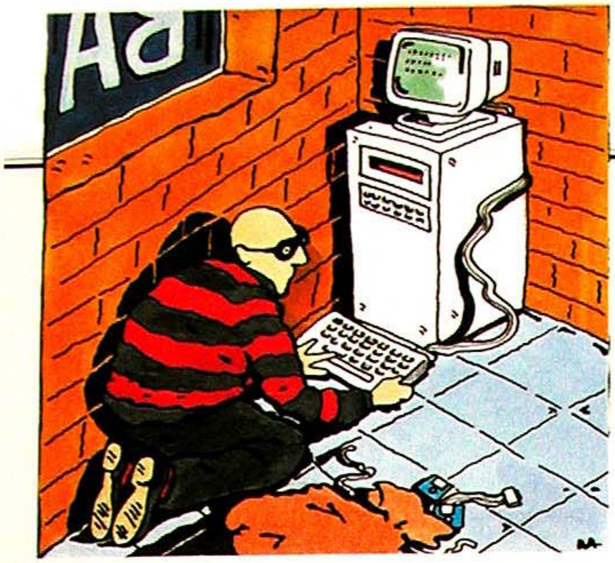

Safeguarding confidential electronic communication has become a highly complex undertaking. Malcolm Hamer looks at the options

KEY TO SECURITY

The banking community is, for obvious reasons, very concerned with the security of electronic movement of information. In the past 30 years the movement of money between and within banks, particularly internationally, has grown tremendously; the demands of the marketplace today can only be met by electronic methods of transmitting payment instructions and other transactions.

Other flows of information related to financial transactions (such as account balance information, letters of credit, and market information) also take place electronically. The growth in the volume of information moved, the speed with which transactions can be executed, and the speed with which stolen information can be acted on by a sophisticated information thief make information security of vital importance in the design of every electronic system within a bank.

Banks face a number of different threats in handling electronic information. These can be categorised as follows:

Theft of private information by a passive tap on a leased circuit. Such a theft would be particularly damaging where this information could be used to undermine the customer's financial interests, for example, by exposing one company's plans for taking over another company.

Fraudulent acts committed by an active tap on a leased circuit. In this case, an apparently valid transaction is inserted into a stream of transactions passing between two computers. Alternatively, a valid transaction can be intercepted and changed. Here, account numbers to be credited and/or the amount of the transaction might be targeted.

Fraudulent acts committed by someone who impersonates a customer electronically by entering payment instructions into an electronic banking system, using the customer's ID and password. Passwords can easily be compromised by a passive tap on a customer's or bank's telephone lines.

There are, of course, other types of threat, such as simple theft of information stored on magnetic disks or tapes by someone entering a bank building illegally.

Cryptographic solutions have been employed increasingly in recent years. Any bank not employing protective measures today would be regarded as recklessly irresponsible and, under the laws of some countries, could be exposing itself to legal action by customers for failure to adequately protect their information.

When discussing solutions, there is sometimes confusion between two complementary cryptographic techniques, namely, data encryption and data authentication. Data encryption or encypherment renders data unintelligible to a potential information thief by transforming one stream of data bits into a different stream of bits by means of a series of mathematical steps known as a mathematical algorithm. The algorithm typically has two inputs: the data stream which is to be encrypted and a fixed string of bits which forms the encryption key. The output of the algorithm is the encrypted data stream.

At the receiving end the process is operated in reverse: the encrypted data now forms one input, the second input is the decrypting key, and the output is the original data.

In most commercial applications the encryption algorithm is published and the security of the process depends entirely on the secrecy of keys. The pair of keys - one at the sending end and one at the receiving end - are known only to the sending and receiving parties. Ideally, of courtse, only the pieces of equipment doing the encrypting and decrypting "know" the keys. The method of handling the keys is such as to avoid any human being seeing or typing them.

In military contexts, the algorithm, as well as the keys, may be kept secret. Experts disagree on whether secret algorithms are a strength or a weakness. While keeping the algorithm

Ib

secret generally makes life more difficult for the enemy - the enemy has to guess both the algorithm and the keys - there is a higher probability that a secret algorithm has an intrinsic weakness that makes it easy to crack. This is because only a few experts have tested each secret algorithm to prove that it is uncrackable. By contrast, the well-known public algorithms are constantly being scrutinised by cryptography professors and students world-wide in the hope that the cracking of a well-respected algorithm will garner not insignificant benefits in terms of reputation.

The most widely used data encryption algorithm in commercial applications is the Data Encryption Standard (DES), as developed jointly by IBM and the US National Bureau of Standards. Despite the alogorithm's status as a US and de facto global standard, the US State Department nonetheless attempts to limit the availability of equipment that implements DES by granting export licences only to banks, US multi-national companies and friendly goverments.

DES is a symmetrical algorithm. Thus the pair of keys is, in fact, a single shared key kept secret from the rest of the world. The key consists of 56 bits (although keys are normally exchanged in the form of 64-bit strings which include eight check bits to help detect accidental corruption of the key in transit).

The second cryptographic technique - data authentication - allows the recipient to verify that data has not been changed in transit and that it has been sent by the person or organisation that it appears to have been sent by. Data authentication is typically carried out by appending to the data file or message an authentication code. Generated by an authentication algorithm which has as its two inputs the entire contents of the file/message and a secret key.

The key arrangements are typically established by prior arrangement between two or more communicating parties. The recipient verifies message authenticity with a complementary algorithm which uses, as its two inputs, the entire file/message and a key (which may be identical to the sender's key, or may be one of a pair of keys, depending on the authentication technique being used). If the output of this algorithm matches the authentication code attached to the file/message then the recipient can be have a high degree of confidence that the message has not been tampered with.

The most widely used commercial data authentication schemes are a public standard approved by the International Standards Organisation - ISO Standard 8730 (also known in the USA as ANSI Standard X9.9) - and an unpublished scheme used by Swift. Also known as standard X9.9 of the American National Standards Institute (ANSI), ISO 8730 uses the DES encryption algorithm as part of the process for generating the message authentication code (MAC) which is appended to a message. Data encryption and data authentication are sometimes confused because, in certain situations, encryption may be used as a "stand in" for authentication. It may also be used as a supplementary level of protection along with authentication.

The techniques of encryption and authentication may be used both internally, within a single bank, or externally between a pair of banks or between a bank and its customers. These are some examples:

When banks send financial transactions to one another via the SWIFT network, each message is authenticated using the SWIFT authentication method. The leased circuit links each bank to SWIFT's switching centres and all of the leased circuit links within the SWIFT network are encrypted using link encryptors to protect the confidentiality of customers' transaction details - and also to give a second layer of protection against fraudulent message modification and insertion.

Most banks encrypt all circuits in their private leased circuit networks to protect the confidentiality of their customers' information and their own internal bank information. Encryption also gives a second layer of protection against message modification and insertion.

Many banks use the ISO 8730 message authentication technique to attach MACs to all transactional messages sent between their branches, thus protecting themselves against message modification and insertion. The MACs are typically generated and checked by means of a so-called box - that is, a piece of hardware separate from the computer system. It is generally considered more secure to carry out the MAC generation and checking process in separate device because this makes it impossible for a computer programmer to interfere with the operation of the MAC process.

Several banks supply their corporate customers with ISO 8730 message authentication boxes to attach to the PCs used to send payment instructions to the bank. Typically, a customer will connect to the bank's computer by a dial-up data call through the local telephone network, and then sign on to the bank's computer with an ID and password. These give a level of protection against a thief impersonating the customer, although this is not considered adequate for high-value payments. The PC is programmed to pass payment instruction messages and other important messages through the ISO 8730 box, which appends a MAC to each message before it goes over the dial-up connection to the bank's computer.

➡

Several banks also strengthen the security of their customers' ID/password sign-on process in one of two ways.

The bank may supply the customer with either a box or a PC software module to encrypt the password using a different key for each session, thus making it impossible (or at least very difficult) for a thief to break into the customer's account by putting a passive tap on the customer's or bank's telephone lines and recording the ID and password.

A password recorded on one session will not work on the next session. Alternatively, the bank may supply the customer with a box that contains a link encryptor that encrypts the entire session using DES encryption. Password or session encryption may be used in combination with message authentication. In this case the bank will generally ask its equipment supplier to combine the authentication function and the encryption function within the same box attached to the PC, thus minimising the cost of the arrangement.

Any bank not employing protective measures can be regarded as recklessly irresponsible. The potential for customer wrath should also be considered.

The existence of tried and tested standards like DES encryption and ISO 8730 authentication may make the whole business of security sound straightforward. However, there is a catch. All these techniques require that the sending end and receiving end of a stream of data or a series of messages reach agreement on the key or keys to be used for encryption and/or authentication and that they do so without anyone else knowing or being able to guess the key or keys. This is the key management problem. Almost all cracking of codes and cyphers, military and commercial, has resulted from sloppy handling of the key management process.

To improve the security of protective measures, it is necessary to eliminate any handling of keys by human beings and, ideally, make the entire key management process automatic. Known in the US as ANSI standard X9.17, ISO 8732 defines a scheme whereby two pieces of equipment can go through an electronic handshake process to agree a key to be used for a particular session or transaction. However, the process depends on the units sharing a set of master keys that must be programmed before the units are installed at their respective sites. Thus, ISO 8732 is extraordinarily difficult to implement in a large global network, particularly if many different banks and their customers are involved.

In the context of message authentication, ISO 8730 allows for the sending party to randomly select a key from a large pool of keys and indicate which key has been selected by means of a key index (for example: "I'm using key number 5769 for the MAC on this message"). However, this means that each pair of would-be communicators has to establish a shared pool of keys by programming the units at a central depot where a highly secure master unit holds all the pools of keys for all the slave units in the world. In practice, there would need to be one depot per country or major city, each containing a master unit with millions of keys already stored in memory ready for assignment to new slave units.

This is just about workable for a single bank and its customers, but is totally impractical for a multi-bank, multi-customer scheme. Of particular concern is the fact that the entire global network of communicating units would be compromised by the theft of one of the master units, necessitating the re-programming of every master unit and every slave unit in the world - a task that could take many months.

Another approach to solving the key management problem is to adopt a cryptographic arrangement which gets away from the need for a single secret key to be shared by the sending and receiving devices. A number of so-called public key schemes based on asymmetrical alogorithms were developed in the late 1970s.

In a public key scheme each user has two keys - a private key and a public key. The two keys are ideally generated in a sealed unit containing memory, a battery, and a processor chip. After the generation of the pair of keys the private key is held in the memory and the public key released from the sealed unit. In this way even the user never sees or touches his or her private key. Once generated, the pair may be safely used for at least a year, provided that the key is long enough. The private key is used to decrypt data received by that user and to generate the MACs on messages sent out by that user to other users.

The user's public key is used by anyone who wants to send encrypted data to the user in question or check that user's MACs. It can be published in a directory because the mathematical relationship between the public key and the private key is so complex that the private key cannot be derived from the public key. When a user's public key is used to encrypt data sent to the user, only that user can decrypt it because only that user holds the private key. The public key can also be used to check the MAC on a message that has been generated by the user. Thus, anyone in the world can check the authenticity of a message by verifying the user's public key in the directory and using it to check the MAC.

The roles of the public and private keys are reversed between authentication and encryption. The public key is used in encrypting messages to its owner and in checking MACs from its owner. The owner uses his or her private key for decrypting incoming messages and in generating MACs for outgoing messages.

Public key schemes are so obviously superior to symmetrical (single key) schemes like DES - for encryption - and ISO 8730 - for authentication - that it may seem surprising that they have not already replaced symmetrical schemes. However, there are two problems. The first problem lies in the realm of high-speed link encryption. The mathematical operations involved in public key cryptography are computationally intensive. Even with ➨

today's very fast microchips, it is impossible to encrypt data at a rate of more than about 300 bits per second. It is not therefore possible to build a link encryptor operating at useful speeds (such as 9600 bits per second) using either of the two well-known public key algorithms (the RSA algorithm and the Diffie-Hellman algorithm).

However, there is no problem in building an authentication device that can generate, in a couple of seconds, a MAC (also known in this context as a digital signature). Similarly, there would be no problem encrypting small amounts of data using public key techniques. Passwords, for instance, could be encrypted using RSA, although it would be necessary to incorporate the hand-over of the password in a challenge/response protocol to provide a random element in each sign-on. This is because the public and private keys do not change with each session.

The second problem concerns authentication using public key techniques. Fairly cautious when it comes to security, many banks are waiting for an international standard to be published before rushing out to buy public-key based equipment and software for message authentication. The normal standards process has been slowed down by lack of interest from the US standards bodies. This is reportedly a result of pressure from the US National Security Agency which is concerned that the publication of a standard for authentication using public key cryptography will soon lead to a standard for public key encryption - which in turn will lead to the manufacture of public key encryption devices, which in turn will be bought by some of the less sophisticated countries on which they wish to spy, which in turn will make their job more difficult.

Whether this theory is true or not is impossible to verify. But in any case, European banks and other organisations are pushing for an ISO standard, so we may for once see an ISO standard in this field which is not simply a US standard with a few editorial changes. Several European banks are already using an authentication scheme for inter-bank transactions based on the RSA algorithm, in anticipation of this becoming a standard.

Even in the absence of fast enough microchips to do link encryption using a public key algorithm it is still possible to enjoy the benefits of the simpler key management process under public key operation, while using DES for the actual encryption process. Several link encryptors are available that use a public key technique to exchange DES keys. For example, Cylink of Sunnyvale, California, manufacture a range of link encryptors that use an application of the Diffie-Hellman algorithm known as SEEK (Secure Electronic Exchange of Keys) to exchange DES keys periodically.

With SEEK, a thief tapping the line can record the key-exchange process but cannot, even with the most powerful computer available today, determine what the DES keys were that were exchanged. Unlike conventional schemes such as ISO 8732, SEEK does not depend on an initial set of master keys being shared between the two ends. Each key-exchange action stands on its own and does not depend on the previous exchange in any way.

Link encryptors like Cylink's have gained rapid acceptance in banks because they are easy to set up and require no manual intervention whatsoever. The key-change process is automatic and can be programmed to occur, say, every night at midnight, or even more frequently if necessary. By contrast, keys are rarely changed more than once a month on link encryptors that require manual intervention for key changes.

Although applications of encryption and authentication are common in the banking community, many of the technical solutions for authentication and session/password encryption are clumsy as a result of lack of good standards. The areas in which standards are most lacking are practical key management and public key authentication (digital signatures).

Although the ISO 8732 key management standard provides a framework for the management of DES keys, there are many details of implementation to be worked out in a practical situation. As a result, only by extensive co-operation could two or more banks achieve a solution where a single box at the customer's office could talk to the systems at more than one bank - and of course, competition between banks makes such co-operation difficult. So, customers who use several banks have to have either one PC for each bank or a number of boxes attached to one PC (and these boxes often interfere with one another's operation).

We may for once see an ISO standard in this field that advances beyond the usual situation of utilising a US standard with a few editorial changes

The only situation in which good solutions for authentication, including key management, have been worked out are where a single body has specified standards in great detail and all banks have had to comply - the SWIFT network being the obvious example. It is a shame that SWIFT, viewed as a standards society as well as a network operator, has not been tasked by its members with the job of agreeing standards for bank/customer communication, particularly as regards security. With the rapid growth in EDI (Electronic Data Interchange), it is vital that international standards for authentication of payment instructions (and other forms of contractual commitment) be worked out internationally in the very near future.

A global scheme based on ISO 8730 is simply unworkable and only public key techniques will meet the needs of EDI. The major banks should be more active in pushing (either directly or through SWIFT) for an early ISO standard for public key based authentication. Indeed, US banks in particular should be more vocal in criticising the lack of US standards organisation participation and anti-public key pressures from US government agencies.

Malcolm Hamer is head of Asia Pacific Regional Telecommunications for Citibank NA.