OSI AND X.400 SECURITY

Chris J Mitchell and Michael Walker

1. INTRODUCTION

The work of the main international standards committees involved in work on security for OSI (i.e. ISO, CCITT and ECMA) can be divided into three main parts. First there is work on underlying techniques, such as: cryptographic algorithms, modes of operation for cryptographic algorithms and peer entity authentication mechanisms. Second there is more general work describing how these techniques may be used to provide security in both OSI applications and various layers of the OSI model, such as: the OSI security architecture, Lower and Upper Layer security models and various security frameworks. Third there is work on specifying how security should be provided in specific OSI applications, as typified by the security elements of the 1988 versions of the CCITT X.400 standards.

In this paper we attempt to do two things. We first survey the past and current efforts on security in OSI in all of the above three areas. Second, we consider in more detail one specific OSI application, namely CCITT X.400 store-and-forward messaging. In particular we consider the security elements provided there, and explain how they might be used to provide secure messaging.

2. OSI SECURITY ACTIVITIES

2.1 Security techniques

Within ISO, work on techniques for security, in particular on cryptographic techniques, has been primarily focussed within ISO/IEC/JTC1 SC20 (and will be continued by its proposed successor SC27). Outside ISO, other work has proceeded within ANSI and the NBS (in the U.S.A.). We divide our consideration of this work into three areas: algorithms (e.g. encryption functions, digital signature functions), peer entity authentication protocols and key management.

2.1.1 Algorithms

Encryption algorithms were the first type of algorithm to receive consideration by ISO. This followed earlier work by ANSI in the U.S. resulting in the adoption of the Data Encryption Standard (DES) block cipher algorithm as a U.S. National Standard, [1], [15]. Considerable efforts were made in the early 1980s to get this same algorithm adopted as an international standard (to be called 'Data Encryption Algorithm 1'). A draft international standard was prepared, ISO DIS 8227, [19], but this failed to obtain approval for adoption as an international standard. Subsequently it was decided that ISO would not attempt to provide any standards for encryption algorithms (i.e. techniques for providing data confidentiality).

Instead, it has been decided to adopt the idea of an international register of algorithms, through which any encryption algorithm can be given a standardised identifier. The draft proposal ISO DP 9979, [31], caters for registering proprietary algorithms, the details of which may remain confidential to their owners.

When the DES was adopted as a standard in the U.S.A., four 'modes of use' were also standardised, [2], [16]. These modes of use specify how the DES encipherment algorithm (a 64-bit block cipher) should be used. This work was also taken up by ISO, resulting in an international standard, ISO 8372, [20], specifying modes of use for an arbitrary 64-bit block cipher algorithm. A successor to this standard, in the form of a draft proposal, ISO DP 10116, [32], generalises this further to specifying modes of use for an N-bit block cipher algorithm. The recent ANSI standard, X9.23, [6], details how the DES algorithm should be used to provide encryption of wholesale financial messages.

In addition to data confidentiality, a good deal of work has also been done within ISO (and other standards bodies) concerning standardising algorithms for message authentication, integrity checking and digital signature. The initial work on message authentication was undertaken by financial groups within ANSI, resulting in ANSI standards X9.9 and X9.19, [3], [5], which give standard methods for authenticating financial messages. Parallel work, again specific to the financial application, has been carried out within ISO, resulting in ISO 8730 and ISO 8731 (parts 1 and 2), [21], [22], [23]. For more general application, a draft international standard now exists, ISO DIS 9797, [28], for a data integrity mechanism.

Finally, two standards proposals exist relating to digital signature algorithms. The first is a draft proposal, ISO DP 10118, [34], specifying possible methods for computing hash functions for digital signatures; note that one of the methods described there will probably need to be removed in the light of recent work by Coppersmith, [12]. The second is a proposal for a signature algorithm for 'short' messages, ISO DP 9796, [27].

2.1.2 Peer entity authentication

In parallel with the current work within ISO on algorithms, efforts have also been made to standardise the protocol exchanges involved in performing party-to-party authentication. This has resulted in three draft proposals for standards, one covering the use of conventional (secret-key) cryptography, ISO DP 9798, [29], and the other two based on the use of public-key cryptography, ISO DPs 9799 and 10117, [30], [33]. The exact form in which these documents should proceed towards standard status remains to be decided; currently there are moves to reduce the number of documents by producing a multi-part standard which would include DPs 9798, 9799 and 10117, as well as including a section on zero-knowledge authentication protocols. ANSI is also active in the area of authentication, and is preparing a standard, X9.26, [8], on access security for wholesale financial systems, which includes secure transmission of personal authenticating information and node authentication.

2.1.3 Key management

ISO work on this topic is at an early stage of development. Three draft documents exist, entitled: Cryptographic mechanisms for key management: Part 1: Key management overview, Part 2: Key management using secret key techniques and Part 3: Key management for public key register. It is likely to be some time before any of these documents emerge as Draft Proposals, since at the moment none of them are any where near completion.

The general ISO work on key management will need to take account of earlier work in this area, in particular that undertaken for the financial community. As with a number of other security standards, ANSI have led the way, with the production of ANSI standards X9.17 and X9.24, [4], [7], specifying how key management should be performed for certain kinds of financial application. Like all ANSI standards, only management of symmetric keys using symmetric techniques is considered. ISO has also been active in this area, with the production of a standard, ISO 8732, [24], again describing symmetric key management for financial applications.

2.2 Using security mechanisms

Within ISO, the questions of how and where within the OSI model security mechanisms are to be used falls primarily within the scope of SC21/WG1, together with the layer and application specific Working Groups of SC6 and SC21. We divide our discussion of this topic into three parts, covering: security architectures and models, security frameworks and layer specific standards.

2.2.1 Security architectures and models

To date, the main achievement in this area has been the production of the OSI Security Architecture, ISO 7498-2, [18], in 1988. This document covers a number of important topics, including: standardised definitions of security terminology and security services, a guide to the relationship between security services and mechanisms, an indication of which security services are relevant to which layers of the OSI model and a short introduction to security management.

Subsequent to the production of this standard, work has started within SC21 and SC6 on two security models: a Lower Layer Model (relevant to OSI Layers 1-4) and an Upper Layer Model (relevant to OSI Layers 5-7). These models are intended as general guides to the insertion of security facilities into the relevant layers of the OSI model. Work on these two models is at an early stage and has not yet reached DP status.

In parallel with these activities, security facilities are under consideration both within the ISO Open Distributed Processing group (SC21/WG7) and the CCITT's Distributed Applications Framework (DAF) activity. Finally we briefly mention the ECMA work in this area. ECMA have produced a technical report entitled Security in Open Systems - A Security Framework, [13] and have also produced a draft for an ECMA standard entitled Security in Open Systems: Data elements and service definitions, [14]. These documents are likely to be most significant in terms of the influence they have over subsequent ISO standards and CCITT Recommendations. They have particular relevance to the provision of access control services in distributed systems.

2.2.2 Security frameworks

Another recently inaugurated work topic within ISO/IEC/JTC1 SC21 covers the 'security frameworks'. This projected six-part standard will give a framework for the provision of particular security services in distributed systems. The six parts will cover the following topics:
Part 1: Authentication Framework

Part 2: Access Control Framework,
Part 3: Non-repudiation Framework
Part 4: Integrity Framework
Part 5: Confidentiality Framework
Part 6: Audit Framework

In addition there will be a Part 0, giving a general introduction to the six security frameworks. All these documents are at an early stage of development, although it is hoped to progress Parts 1 and 2 to DP status within the next few months.

2.2.3 Layer specific standards

Apart from the Upper and Lower Layer Security Models, a number of other drafts are in existence covering the provision of security services in specific layers of the OSI model. An ISO standard exists, ISO 9160, [25], specifying how security should be provided in Layer 1 (Physical). Within IEEE 802.10, work is progressing on standardising the provision of security in LANs, [17]; the proposed security functionality all resides in Layer 2 (Link). In the U.S. SDNS activity, work is progressing on two documents, [36], [37], specifying how security should be provided in Layers 3 (Network) and 4 (Transport). During the past few years ISO/IEC/JTC1 SC20/WG3 has also produced draft documents relating to the provision of security in Layers 3, 4 and 6 (Presentation), although the future of that work is not clear at the moment.

2.3 Security for OSI applications

We now very briefly consider the effort that has been devoted to providing standardised security solutions for specific OSI applications. Then in section 3 we consider in more detail the security provisions made for one particular OSI application, namely X.400 electronic mail.

2.3.1 X.500 security

The 1988 version of the X.500 CCITT Recommendations on Directory Services, [11], and their corresponding ISO draft standards, [26], include means to use the Directory Service to provide key management and peer-entity authentication through storage of user public keys in the directory. The 1992 version of these recommendations is also expected to contain detailed provisions for access control to directory entries.

2.3.2 X.400-1988 security

The 1988 versions of the X.400 CCITT Recommendations, [10], include a variety of security features making it possible to provide a variety of security services for electronic mail. We discuss these provisions in more detail in the next section.

3. SECURITY FOR X.400 STORE-AND-FORWARD MESSAGING

3.1 Introduction

We now consider in more detail security in electronic mail applications, with particular reference to the security features in the 1988 versions of the CCITT X.400 Recommendations, [10]. We devote the remainder of

this introduction to a brief review of the fundamental concepts underlying the X.400 electronic mail system.

The 1984 version of the X.400 recommendations, [9], define two basic types of entity in a 'store and forward' mail network, namely User Agents (UAs) and Message Transfer Agents (MTAs). UAs originate and receive messages on behalf of users. All messages are sent via one or more MTAs, which act as 'store and forward' message nodes. The set of all MTAs collectively form what is known as the Message Transfer Service (MTS).

X.400 is widely used as a generic term for a collection of related C.C.I.T.T. Recommendations, including X.400 itself, X.402, X.411, X.413 and X.420, [10]. The protocols governing communication between pairs of MTAs and between a UA and the MTS are defined in X.411. The protocol governing MTAÄMTA communications is often referred to as P1, and the UAÄMTA protocol as P3. The entire collection of UAs and MTAs is referred to as the Message Handling System (MHS).

In the 1988 version of the X.400 Recommendations, [10], in fact in X.413, a third type of entity is defined, namely a Message Store (MS). Message Stores were not part of the 1984 version of X.400. In some cases it is convenient to only connect a UA to the MTS at very infrequent intervals. However MTAs may only store mail for recipient UAs for a short period of time. The role of a MS is to remedy this problem by acting as an intermediary between a UA and the MTS, with storage of received messages as its primary role. UAs and MSs are in 1Ä1 correspondence, and an MS enables its corresponding UA to obtain summary information about received messages without actually retrieving them. In practice, an MS is likely to be co-located either with an MTA or with its corresponding UA. The Message Store Access Protocol, governing the retrieval of messages by a UA from its corresponding MS, is defined in Recommendation X.413. Note that UAs and MSs are collectively referred to as MTS-users, in that they are both end-users of the Message Transfer Service.

All the protocols so far discussed, namely those in X.411 and X.413, have the role of defining how an object called a message-content is shipped from one UA to another. The form of this content is not constrained by X.411 or X.413, and may be one of a number of different types. It is carried transparently by the MTS. One such type is defined in X.420; this type is defined as suitable for use in Inter-Personal Messaging applications. Other content types may be defined for different applications such as Electronic Data Interchange (EDI).

Finally note that the set of parameters defined in X.411 and X.413, which accompany the message content when it is transferred from one MHS entity to another, are often referred to as the message envelope. This is because in many ways these parameters have roles analogous to those of the addressing and franking information to be found on the envelope of conventional paper mail. However, in X.400 the form and content of the envelope depends on the type of entities which are involved in the transfer, e.g. submission envelopes are used to transfer from a UA to the MTS and delivery envelopes are used to transfer from the MTS to a UA.

3.2 Security services

Before describing electronic mail security services in detail, it is useful to consider what threats these services are intended to counter.

Possible threats to electronic mail systems include: masquerade, message replay/re-sequencing, modification of message information, denial of service, leakage of information and repudiation. It is not possible to address all these threats from within a message handling application. For example information leakage will take place if it is possible to monitor the volumes of traffic going from one point in the network to another, even if all the message contents are encrypted. To prevent this requires the provision of security services in the lower layers of the OSI stack, which is beyond the scope of application services.

There are a considerable number of different security services that could be provided within an electronic mail system. Such services may conveniently be divided into two classes, namely MTS-user to MTS-user services and MTS services (note that this is non-standard terminology).

MTS-user to MTS-user services are those provided from one MTSÄuser (i.e. a UA or an MS) to another, without active participation by the MTS. Such services include: Message origin authentication, Proof of delivery, Content confidentiality, Content integrity, Message sequence integrity and Non-repudiation services.

MTS security services are those provided which involve active participation by the MTS. Such services include: Secure access control to the MTS and between MTAs, Report origin authentication, Probe origin authentication, Proof of submission, Non-repudiation of submission and Message security labelling.

The service names used here are those given in the X.400 Recommendations. These do not correspond precisely with the names used in ISO 7498-2, the OSI security architecture, [18]. This is partly because the OSI security architecture does not mention all the services relevant to electronic mail, and partly because the documents were developed in parallel.

3.3 Approaches to providing security

In order to provide security services for the message content it is normally necessary to transmit with the message a number of 'security parameters', e.g. encrypted keys and authentication checks. These security parameters can either be transmitted in the message envelope or as part of a (specially formatted) message content, or both. The choice of location for the security parameters not only has important system ramifications, but can also affect the type of security service which may be provided.

If security services are required for X.400-1984, or other electronic mail systems without built in security features, then there is no alternative but to put the security parameters in the message content. The same is true for any heterogeneous mail systems, even if they individually incorporate security features. Examples of electronic mail systems in which all the security parameters are in the message content are provided by the SDNS and IAB Internet mail security proposals. However, security parameters within the message content cannot be used to provide MTS security services.

A distinct feature of the 1988 X.400 Recommendations is that the message envelope is used to transfer security parameters, and not the message content. The inclusion of the security parameters in the message

envelope enables the provision of MTS security services. However, it does make the provision of certain MTS-user to MTS-user services rather problematical, especially if Message Stores are used.

3.4 Security mechanisms

Before we consider the security mechanisms described in the X.400 Recommendations, we need to consider the provision of cryptographic key management, a fundamental requirement for the provision of communications security services. Key management for the X.400 security facilities is achieved by use of the directory authentication service specified in C.C.I.T.T. Recommendation X.509, [11]. This key management system is based on the use of public key cryptosystems for digital signature and data encryption. Recommendation X.509, [11], allows public keys to be stored in user directory entries.

Since the directory service (and communications with it) may not be trusted, means need to be provided for users to verify public keys read from the directory. This is provided for by the use of data structures called certificates, which we now briefly describe.

In order to set up a key management system for X.400, every user who wants to use security services must first exchange public keys with an off-line entity called a Certification Authority (CA). Each user must trust the CA which they appoint to act on their behalf. The CA gives the user a copy of its public key (each CA has its own public key/secret key pair), and is given in return a copy of the user's public key (each user must also equip themselves with a key pair). The CA then signs a copy of the user's public key, together with the user's name and the period of validity of the key, using the CA's secret key. This forms a certificate and is actually what is put in the directory. Any other user which has a trusted copy of this CA's public key can then check the validity of the certificate, and thereby obtain a verified copy of the user's public key.

The scheme so far described does not cover the situation where two users are served by different CAs. To cover this possibility, one CA may generate a certificate for another CA's public key; such certificates are called 'cross-certificates'. If user A has CA X, and user B has CA Y, then if A is given a cross-certificate containing Y's public key signed by X, then A can obtain a verified copy of Y's public key. Once it has this key, A can then check B's certificate. Such cross-certificates can be made into chains called 'certification paths'.

Virtually all the security services built into the X.400 Recommendations make use of a cryptographic construct called a token. Tokens are always formed for a single recipient. A token consists of a series of data fields with a digital signature appended, this signature being computed as a function of all the data fields in the token (using the originator's secret key). These data fields include: recipient-name, date/time of generation, a field called 'signed-data' and a field called 'encrypted-data'. The information within the encrypted-data field is enciphered using the public key of the intended recipient of the token (prior to computation of the signature).

One form of token is called a message-token, and is used in the provision of all the MTS-user to MTS-user security services. Hence, if a message requires such services, then a message-token is sent as one parameter

within the message envelope. The precise contents of the signed-data and encrypted-data fields within the message-token depend on which selection of security services is required. However, whichever services are required, the presence of these data within the token prevents them from being changed and/or repudiated.

In a message-token, the encrypted-data field may be used to contain any of the following items: a cryptographic key (used to encrypt the message content if content confidentiality is required), a content integrity check (used in the provision of content integrity), a message security label, a content integrity key (used to compute the content integrity check) and a message sequence number (used in the provision of message sequence integrity). The signed-data field may be used to contain any of the following items: a content integrity check (used in the provision of content integrity), a message security label, a message sequence number (used in the provision of message sequence integrity) and a proof of delivery request.

The proof of delivery and non-repudiation of delivery services are slightly different from other MTS-user to MTS-user services in that they are provided by the message recipient to the message originator. If a message is received containing a proof of delivery request (in the signed-data field of the message token) then the recipient computes a signed version of the (unencrypted) message content together with other delivery related parameters. This signature, computed using the recipient's secret key, is returned to the message originator within the delivery report. The message originator then uses this signature to provide the required service(s).

Means are also provided within X.411 and X.413 for a pair of MHS entities to perform peer-entity-authentication prior to opening a connection for the exchange of messages. This protocol exchange again involves the use of tokens. For systems providing Mandatory Access Control services, all messages and entities can be assigned security labels. These labels can be tied to message contents by their inclusion in either the encrypted-data or signed-data fields of the message token (depending on whether or not the label itself is confidential). Inter-entity connections can also be assigned security-labels using the tokens exchanged in the peer-entity-authentication process.

3.5 Limitations of security in X.400-1988

We conclude by very briefly mentioning three important limitations of the current X.400 Recommendations. A more detailed discussion of these shortcomings can be found in [35].

First, proof of delivery to a UA is not available when an MS is used. Because of the way the protocols operate, the proof of delivery must be generated at the time the message is delivered by the MTS to the MTS-user. If this MTS-user is an MS, then it must generate and sign the delivery proof, and not the end user. The message originator then has no proof that the message was ever delivered to the recipient UA, only to the MS belonging to the recipient UA.

Second, proof of delivery by an MS is not possible if the message content is encrypted. The proof of delivery must be computed using the

unencrypted message content, which will not be available to the MS (unless the MS is equipped with the UA's secret key).

Third, the specified form of token may allow the 'theft' of message content by third parties. This arises because the signature on the token is computed after the secret data (in the encrypted-data field) is enciphered. The problem would not arise if the order of these two operations was reversed.

REFERENCES

[1] ANSI X3.92-1981, Data encryption algorithm, American National Standards Institute (New York), 1981.

[2] ANSI X3.106-1983, American National Standard for Information Systems - Data Encryption Algorithm - Modes of Operation, American National Standards Institute, New York, 1983.

[3] ANSI X9.9, Financial institution message authentication (wholesale), American Bankers Association, Washington, DC, August 1986.

[4] ANSI X9.17, Financial institution key management (wholesale), American Bankers Association, Washington, DC, April 1985.

[5] ANSI X9.19, Financial institution retail message authentication, American Bankers Association, Washington, DC.

[6] ANSI X9.23, Financial institution encryption of wholesale financial messages, American Bankers Association, Washington, DC, 1988.

[7] ANSI X9.24, Financial services - retail key management, American Bankers Association, Washington, DC, Draft 5.0, 1987.

[8] ANSI X9.26, Access security for wholesale financial systems: Secure transmission of personal authenticating information and node authentication, American Bankers Association, Washington, DC, Draft 5.0, 1988.

[9] C.C.I.T.T. Recommendations X.400, X.401, X.408, X.409, X.410, X.411, X.420, X.430, C.C.I.T.T. VIIIth Plenary Assembly, October 1984.

[10] C.C.I.T.T. Recommendations X.400, X.402, X.407, X.411, X.413, X.419, X.420, Message handling systems, C.C.I.T.T. IXth Plenary Assembly, October 1988.

[11] C.C.I.T.T. Recommendations X.500, X.501, X.509, X.511, X.518, X.519, X.520, X.521, The Directory, C.C.I.T.T. IXth Plenary Assembly, October 1988.

[12] D. Coppersmith, Analysis of ISO/CCITT Document X.509 Annex D, preprint, IBM Thomas J Watson Research Center, 1989.

[13] ECMA TR/46, Security in open systems - A security framework, ECMA, July 1988.

[14] ECMA Draft Standard, Security in open systems - Data elements and service definitions, Output of the 11th (Bristol) meeting, ECMA/TC32/TG9, June 1989.

[15] FIPS PUB 46, Data encryption standard, Federal Information Processing Standards Publication 46, National Bureau of Standards, U.S. Department of Commerce, Washington, DC, January 1977.

[16] FIPS PUB 81, DES modes of operation, Federal Information Processing Standards Publication 81, National Bureau of Standards, U.S. Department of Commerce, Washington, DC, December 1980.

[17] IEEE P802.10/D5, Standard for Interoperable Local Area Network (LAN) Security (SILS), Draft of June 26th 1989.

[18] ISO 7498-2, Information processing systems - Open systems interconnection - Reference Model - Part 2: Security Architecture, International Organization for Standardization, 1988.

[19] ISO/DIS 8227, Information processing - Data encipherment - Specification of algorithm DEA 1, International Organization for Standardization, 1985.

[20] ISO 8372, Information processing - Modes of operation for a 64-bit block cipher algorithm, International Organization for Standardization, 1987.

[21] ISO 8730, Banking - Requirements for message authentication (wholesale), International Organization for Standardization, 1986.

[22] ISO 8731/1, Banking - Approved algorithm for message authentication - Part 1: DEA, International Organization for Standardization, 1987.

[23] ISO 8731/2, Banking - Approved algorithm for message authentication - Part 2: Message authenticator algorithm, International Organization for Standardization, 1987.

[24] ISO 8732, Banking - Key management (wholesale), International Organization for Standardization, 1988.

[25] ISO 9160, Information processing - Data encipherment - Physical layer interoperability requirements, International Organization for Standardization, 1987.

[26] ISO/DIS 9594-1, 9594-2, 9594-3, 9594-4, 9594-5, 9594Ä6, 9594-7, 9594-8, Information Processing Systems - Open systems interconnection - The Directory, International Organization for Standardization, 1988.

[27] ISO/3rd DP 9796, Data cryptographic techniques - Digital signature scheme giving message recovery, International Organization for Standardization, 1989.

[28] ISO/DIS 9797, Data cryptographic techniques - Data integrity mechanism using a cryptographic check function employing an n-bit algorithm with truncation, International Organization for Standardization, 1988.

[29] ISO/DP 9798, Peer entity authentication mechanisms using an n-bit secret-key algorithm, International Organization for Standardization, 1988.

[30] ISO/3rd DP 9799, Peer entity authentication mechanisms using a public-key algorithm with a two-way handshake, International Organization for Standardization, 1988.

[31] ISO/DIS 9979, Data cryptographic techniques - procedures for the registration of cryptographic algorithms, International Organization for Standardization, 1988.

[32] ISO/DP 10116, Information processing - Modes of operation for an N-bit block cipher algorithm, International Organization for Standardization, 1988.

[33] ISO/DP 10117, Peer entity authentication mechanisms using a public-key algorithm with a three-way handshake, International Organization for Standardization, 1988.

[34] ISO/2nd DP 10118, Information Technology - Data encryption - Hash-functions for digital signatures, International Organization for Standardization, 1989.

[35] C. Mitchell, D. Rush and M. Walker, 'CCITT/ISO standards for secure message handling', IEEE Journal on Selected Areas in Communications 7 (1989) 517-524.

[36] SDN.301, SDNS Secure Data Network System, Security Protocol 3 (SP3), Revision 1.5, 15th May 1989.

[37] SDN.401, SDNS Secure Data Network System, Security Protocol 4 (SP4), Revision 1.3, 2nd May 1989.