# THE SECURITY ISSUE

Chris J Mitchell

## Foreword

This paper is not intended to be a complete overview of its subject area. It is being prepared as a written accompaniment to a talk to be given at the Value Added Network Services for Europe Conference, due to take place on September 12th and 13th 1989 in London. All presenters of papers at this conference have been requested to provide a 1500-word synopsis of their talk for circulation to attendees at the conference.

## 1. Introduction

The purpose of this paper is to describe security in electronic mail applications, with particular reference to the security facilities in the 1988 versions of the CCITT X.400 Recommendations, [2]. The two main objectives of this paper are first to provide tutorial information on the security facilities within the 1988 X.400 Recommendations and second to point out potential shortcomings in the protocols and areas where improvements can be made.

We devote the remainder of this introduction to a brief review of the fundamental concepts underlying electronic mail systems. In doing so we use the X.400 model and terminology, which can be applied to a variety of other electronic mail systems.

The 1984 version of the X.400 recommendations, [1], defines 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, [2]. 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, 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 1-1 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. Its value is not affected 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 will 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 a piece of 'conventional' mail.

## 2. Security services

Before describing electronic mail security services in detail, it is useful to consider what threats these services are intended to address. 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 intervention 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 intervention by the MTS. Such services include: Secure access control, 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 which do not correspond precisely with the names used in ISO 7498-2, the OSI security architecture, [3]. 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. 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 facilities, 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.

On the other hand, the 1988 X.400 Recommendations use the message envelope 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.

## 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 provided by use of the directory authentication service specified in C.C.I.T.T. Recommendation X.509. This key management system is based on the use of public key cryptosystems for digital signature and data encryption. Recommendation X.509 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. 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.

## 5. Limitations of security in X.400-1988

We conclude this paper by very briefly mentioning three important limitations of the current X.400 Recommendations.

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] 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.

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

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