DRAFT - Technical Report CSL/SNA — 92/?
September 23, 1992

X.400 Security

Paul Markovitz

U. S. Department of Commerce
National Institute of Standards and Technology
Computer Systems Laboratory
Systems and Network Architecture Division
Gaithersburg, MD 20899, USA

Contents

List of Figures

Chapter 1

Introduction

In 1984, the CCITT (Consultative Committee on International Telegraphy and Telephony) approved the first version of the X.400 series of Recommendations. The Recommendations defined a general purpose, store-and-forward, messaging service. In 1988, the CCITT updated the Recommendations to include, among other features, security services that protect messages against modification and disclosure, and allow communicating parties to authenticate their identities.

This paper provides tutorial information about the 1988 X.400 [1] security services. Chapter 1 introduces the paper. Chapter 2 overviews the MHS (Message Handling System). Chapter 3 describes its primary vulnerabilities. Chapter 4 describes the means by which security-relevant information is conveyed in the MHS. Chapter 5 details the X.400 security services that counter the vulnerabilities described in Chapter 3, and Chapter 6 concludes the paper by discussing limitations in the 1988 X.400 security architecture.

Chapter 2

X.400 Overview

This chapter provides an overview of the MHS (Message Handling System). The main aspects of the MHS requiring introduction are: the function model, the message structure, and delivery reporting.

2.1 Functional Model

A functional model of the MHS is shown in figure 2.1. The MHS is a collection of MTAs (Message Transfer Agents), MSs (Message Stores), UAs (User Agents), and AUs (Access Units). MTAs perform the store-and-forward message transfer function. MSs provide storage for messages. UAs enable users to access the MHS, and AUs provide links to other communication systems (e.g., the postal system). A more detailed description of each of these entities follows.

MTAs comprise the MTS (Message Transfer System), the principal component of the MHS. A message is submitted to an MTA by an originating UA, MS or AU, transferred to the recipient MTA(s), and delivered to one or more recipient UAs, MSs, or AUs. If the message is addressed to multiple recipients, the appropriate MTAs perform any splitting (i.e., replicating) of the message needed for delivery to each recipient.

Messages are transferred between MTAs on a cooperating store-and-forward basis. Since no end-to-end association is required, the MTA serving the message recipient need not be active when the message leaves the originating MTA. The message may be stored at a relay (i.e., intermediate) MTA until the recipient MTA becomes operational.

MTAs transfer messages whose content may be encoded in any format. MTAs neither examine nor modify the content of messages except when performing conversion. Conversion increases the effectiveness of the MHS by allowing users to submit messages in one encoded format (e.g., telex), and have them delivered in another encoded format (e.g., IA5). A UA can register with the MTA the encoded information types that may be delivered, and request the MTA to perform any required conversions.

The UA is the MHS component that enables a user to access the MHS, for both the origination and reception of messages. When submitting messages, the UA supplies to an MTA, either directly or indirectly via an MS, the message content, the address(es) of the message recipient(s), and the MTS services being requested. The message content is the information that the originator wants

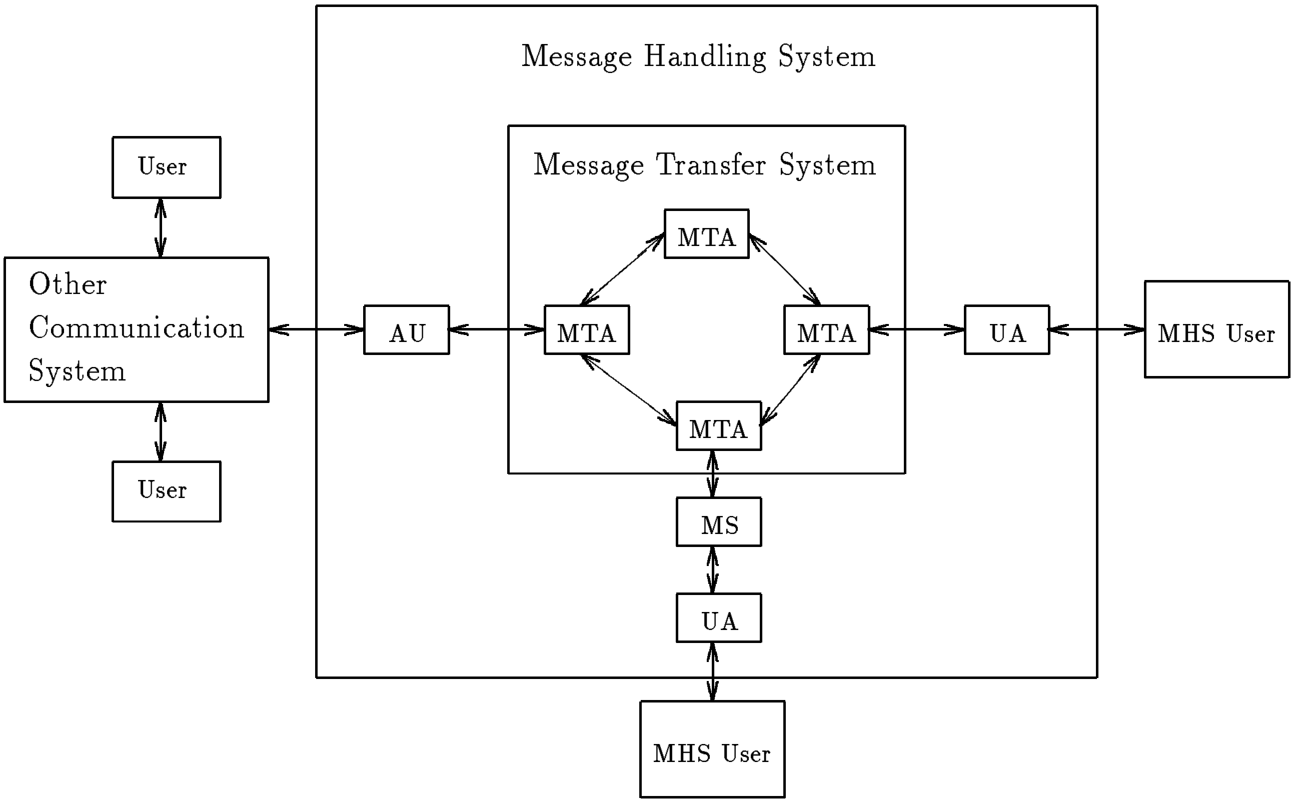

Figure 2.1: MHS Functional Model

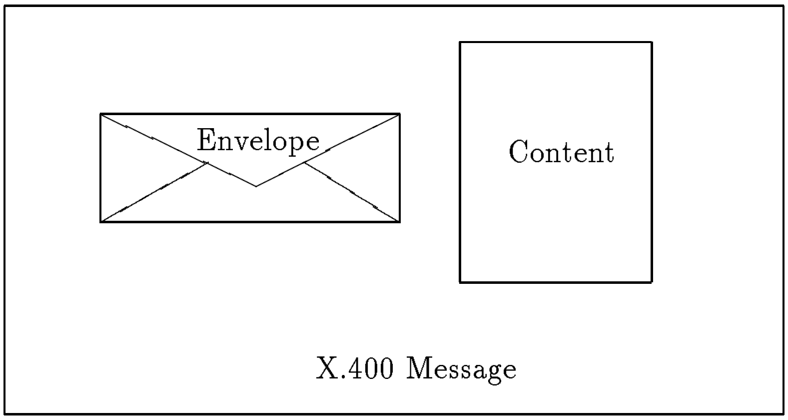

Figure 2.2: MHS Message Structure

transferred to the message recipient(s). The address and service request data are used by the MTS to deliver the message. When receiving messages, the UA may accept delivery of messages directly from an MTA, or it may employ an MS to accept delivery of messages, and retrieve them from the MS at a later time.

The MS is an optional MHS component that acts as an intermediary between a UA and MTA. The MS often co-resides with the MTA serving it. The primary purpose of the MS is to provide a repository for the delivery of messages. The UA can retrieve messages from this repository. By using an MS to accept delivery of messages, a UA is not required to be constantly available. This is especially useful for UA applications implemented on personal computers, which are typically turned off at night. The MS may also submit and forward messages on behalf of the UA, and notify the UA at the time of message delivery.

The AU is the MHS component that provides a gateway between the MHS and another communication system. AUs may, for example, provide intercommunication with telex, teletex, and facsimile systems. Another AU, the PDAU (Physical Delivery Access Unit) enables MHS users to send messages to users residing on a physical delivery system, such as the Postal Service.

2.2 Mesage Structure

The structure of an MHS message is shown in figure 2.2. It consists of a message envelope and a message content. As with a postal message, the envelope represents the information required by the MTS to deliver the message, such as the address(es) of the recipient(s) and any special handling instructions. Many X.400 security parameters are transferred on the message envelope. The message content represents the information that the originator wants conveyed to the message recipient(s).

2.3 Delivery Reporting

The basic X.400 messaging service provides notification of message non-delivery. When a message cannot be delivered to a recipient, a non-delivery report is generated and returned to the originator.

The content of the non-delivery report contains status information about the subject message.

The MT service also provides notification of delivery as an optional service. If a message originator requests acknowledgement of successful delivery, a delivery report is returned to the originator by the recipient's MTA upon delivery of the subject message.

Chapter 3

Vulnerabilities

The distributed nature of the MHS makes it vulnerable to various types of security threats. This chapter describes the nature of these threats, and concludes with a table that correlates each threat with the MHS security service that counters it.

The principal threats to the MHS can be divided into two categories, inter-message threats and intra-message threats. Inter-message threats arise from parties external to the message communication, and include: masquerade, message modification, message sequencing threats, and leakage of information.

Masquerade occurs when an entity successfully pretends to be a different entity. The following examples illustrate two types of masquerade to which the MHS is vulnerable. First, an unauthorized UA may impersonate an authorized UA to gain access to an MTA. Once access is gained, the unauthorized UA can falsely originate messages, falsely acknowledge receipt of messages, or simply discard messages. Second, an unauthorized MTA can impersonate an authorized MTA to misroute messages, or discard messages submitted for delivery.

Message modification occurs when a message is changed by an unauthorized party. Unauthorized changes apply to the message content, addressing information, security labels, and other message attributes. This threat also includes the destruction of an entire message.

Message sequencing threats jeopardize the ordering of messages. They include the reordering, time-shifting (i.e., delaying or pre-playing), and replaying of messages.

Leakage of information occurs when an unauthorized party gains information by monitoring transmissions. The unauthorized party can learn of the content of messages, or of the parties involved in the message transfers. An unauthorized party can also gain information by analyzing the message traffic between two users.

The second category of threats is intra-message threats. Intra-message threats are those performed by the parties involved in the message communication. Intra-message threats include repudiation and security level violations.

Repudiation occurs when an MTA or MTS user (i.e., a User Agent, Message Store, or Access Unit) denies performing a specific action. Repudiation threats include an MTS user denying the origination or delivery of a message, and an MTA denying the submission of a message.

Security level violations are threats relating to security labels. Security labels are data structures which permit the classification of a message, or a communicating party within the MHS, in terms of a security level (e.g., 'Secret'). An example of a security level violation is an originator submitting a message with a security label that it is not authorized to generate.

The threats described above are reproduced below in tabular format. Associated with each threat is the MHS security service or services which counter it. These services are described in detail in Chapter 5.

Masquerade	
Impersonation and misuse of the MTS	Message origin authentication Secure access management
Falsely acknowledge receipt	Proof of delivery
Falsely claim to originate a message	Message origin authentication
Impersonation of an MTA to an MTS user	Proof of submission Report origin authentication Secure access management
Message Sequencing	
Replay of messages	Message sequence integrity
Reordering of messages	Message sequence integrity
Pre-play of messages	
Delay of messages	
Modification of information	
Modification of messages	Content Integrity
Destruction of messages	Message sequence integrity
Leakage of information	
Loss of confidentiality	Content confidentiality
Loss of anonymity	Message flow confidentiality
Traffic analysis	Message flow confidentiality
Repudiation	
Denial of origin	Non-repudiation of origin
Denial of submission	Non-repudiation of submission
Denial of delivery	Non-repudiation of delivery
Security level violations	
Originator not cleared for security label	Secure access management Message security labelling
MTA/MTS user not cleared for security context	Secure access management
Misrouting	Secure access management Message security labelling

Chapter 4

Security-relevant Data Structures

Before describing how X.400 security services counter the threats presented in Chapter 3, three data structures must be discussed. These data structures: the security label, the asymmetric token, and the public key certificate, are used to convey security-related information between communicating parties. This chapter only defines the principal attributes comprising the structures; it provides no details regarding how these structures are used by X.400 security services.

4.1 Security Label

A security label is a collection of attributes associated with an MHS message or entity which permits its classification in terms of a security level. The security label attributes include:

a security policy identifier which identifies the security policy with which the security label is associated,

a printable privacy mark which identifies the level of privacy to be afforded a message or report (e.g., 'In Confidence', 'In Strictest Confidence'),

a security classification which classifies a message or report for security purposes (e.g., 'Unclassified', 'Confidential', 'Top Secret'),

a set of security categories which restricts the context of the privacy mark, the security classification, or both. The categories are application-defined, and may include codewords or caveats to the privacy mark or security classification (e.g., 'Personal-', 'Staff-', 'Commercial-').

Security labels may be transferred in MHS messages and reports, conveyed during the association establishment between two MHS entities (e.g., a UA may transfer security labels when connecting to its MTA), or registered with MHS entities (e.g., an MTA may maintain a registry of security labels for its users).

4.2 Asymmetric Token

The asymmetric token is a signed data structure used to convey security-related information from an originator to a recipient. The attributes comprising the token include:

- the token signature

- the object identifier of the algorithm used to compute the signature,
- any input parameters (other than the token being signed) used to compute the signature,
- the name of the recipient,
- the date and time the token was generated,
- a non-repeating number,
- a collection of fields that is signed (signed-data):
  - a content integrity check,
  - a message security label,
  - a request for proof of delivery,
  - a message sequence number,
- a collection of fields that is encrypted (encrypted-data):
  - a symmetric key used to encrypt the content,
  - a symmetric key used to compute a content integrity check,
  - a content integrity check,
  - a message security label,
  - a message sequence number.

The asymmetric token provides three forms of cryptographic protection. First, it ensures that only the recipient can view the plaintext information in the encrypted-data. This is because the token originator encrypts the encrypted-data using the recipient's public asymmetric key. Thus, only the recipient's private asymmetric key can be used to decrypt the information. Second, it ensures that the token has not been modified. This is because the originator signs the token. The recipient can validate the signature and confirm the token's integrity. Third, it authenticates the identity of the token originator. This is because the originator signs the token using its private asymmetric key. If the recipient validates the signature using the originator's public key certificate, only the originator's corresponding private key could have generated the signature.

X.400 defines two purposes for asymmetric tokens. They can be transferred as credentials when an MHS entity initiates a connection to a peer, and wants to provide strong authentication information. For this purpose the token is referenced as a *bind* token. Tokens can also be transferred in MHS messages, such that a distinct token can be generated for each message recipient. For this purpose the token is referenced as a *message* token.

4.3 Public Key Certificates

In order for a public key scheme to be successful, users must be guaranteed that the public key of another user truly belongs to that user. The means by which a public key is bound to a user is the public key certificate. The public key certificate is a collection of information issued and signed by a CA (Certification Authority). A certificate contains:

- the owner's public asymmetric encryption key,

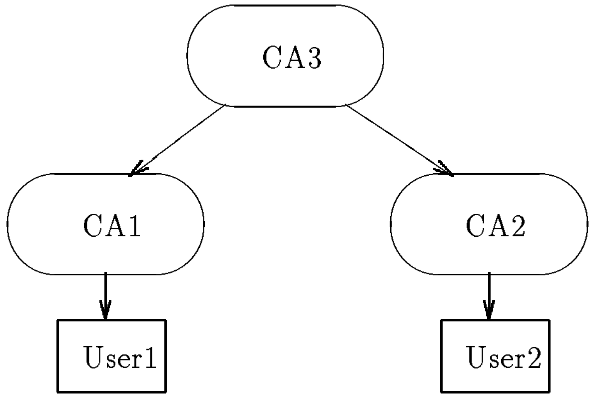

Figure 4.1: Hierarchical Model for Certification Authorities

- the DN (Distinguished Name) of the owner,
- the DN of the certification authority,
- the period during which the certificate is valid,
- a version number,
- a serial number,
- the certificate signature.

When a CA issues a certificate to a user, the CA provides a copy of its public key. With this key the user can validate certificates issued to other users subscribing to the same authority. To validate certificates issued by different authorities, a *certification path* (i.e., a path of trusted certificates) must be constructed between the two users. For example, if *User1* subscribes to *CA1*, *User2* subscribes to *CA2*, and both *CA1* and *CA2* have a trusted relationship with *CA3*, the certification path *(C1, C3, C2)* can be constructed, allowing *User1* and *User2* to obtain each other's public key. This concept of CAs generating certificates for other CAs is called *cross certification.* Figure 4.1 illustrates a hierarchical CA structure for cross certification. In the hierarchical model, CAs only generate certificates for the entities (i.e., CAs or users) below them.

An asymmetric key management scheme is presented in the directory system authentication framework, described in Recommendation X.509 [4]. The directory can be used to store public key certificates for MHS entities. These certificates can be accessed by other MHS entities to compute and/or validate the integrity and confidentiality of MHS messages.

Within the MHS, public key certificates may be conveyed by several methods. When an MHS entity initiates a connection to a peer, it may transfer its public key certificate for the peer to use to validate its credentials. Certificates can be registered with MHS entities, and transferred in MHS messages and reports. MHS entities can also obtain public key certificates by some means outside the MHS, such as accessing the X.500 directory.

Chapter 5

X.400 Services

This chapter presents the MHS security features that counter the vulnerabilities described in Chapter 3. These features can be categorized into three general services. First, they allow various MHS entities to authenticate their identity. Second, they protect messages against modification, and third, they protect messages against unauthorized disclosure.

The security features provided by the MHS apply only to messages submitted directly to an MTA by an MTS user (i.e., a User Agent, Message Store, or Access Unit). They do not apply to communication between the MHS user (e.g., a person) and the MHS (e.g., the person's UA). Thus, the scope of MHS security services extends, for example, to communication between two UAs, but not to communication between two people.

Many of the MHS security services require security capabilities within the UA, but not the MTA. For example, to ensure the confidentiality of a message, the originating UA encrypts the message content, and submits the message to the MTS. Any MTA that handles the message uses envelope information to make decisions (e.g., routing), never needing access to the message content. Some security services, however, require MTAs with security capabilities. For example, to ensure that an originating MTA submitted a message for delivery, the originating MTA must generate and return proof of the submission to the originating user. Some of the MHS security services apply to the MS (Message Store) as well as to UAs and MTAs, such as services involving the exchange of security labels. In general, however, the MS is transparent to security features that apply between the originating and recipient UAs.

Many of the MHS security services rely on encryption techniques. Most services are flexible regarding whether asymmetric or symmetric encryption techniques are used, and more specifically, which algorithms are used. Some services, such as the non-repudiation services, require an asymmetric encryption algorithm.

The remainder of this chapter describes specific MHS security services. These services include: *message security labelling, secure access management, origin authentication, data integrity, data confidentiality, non-repudiation,* and *security management.*

5.1 Message Security Labelling

The *message security labelling* service binds a security label to an MHS object. Security labels can be bound to transferable objects (e.g., messages, reports), MHS entities (e.g., MTAs, UAs), and associations between peer MHS entities (e.g., UAs and MTAs, MTAs and MTAs).

The following simple example illustrates a use of security labels. An MTA services two users: *User1* and *User2*. Each user registers a set of security labels with the MTA. At some later time, a message addressed to both users arrives at the MTA. The message contains a security label with a security classification of *Secret*. The MTA examines the security labels registered for its users, and ascertains that *User1* has registered a *Secret* security label, however, *User2* has not. Depending on the security policy in force, the MTA may deliver the message to *User1*, and non-deliver the message to *User2*. By the same mechanism, the MTA may prevent *User2* from originating messages containing security labels classified as *Secret*.

As illustrated in the example, security labels can be used to control the sensitivity of messages originated by and delivered to a user. Section 5.2.2 describes how security labels can also be used to prevent the misrouting of messages.

5.2 Secure Access Management

Secure access management enables authentication between peer entities in the MHS. It counters the threats of masquerade, misrouting, and the replay of connection requests. *Secure access management* can be divided into two components: *peer entity authentication* and *security context*.

5.2.1 Peer Entity Authentication

The *peer entity authentication* service allows two adjacent components in the MHS to create a secure association by transferring authentication credentials. For example, a UA may provide a password to its MTA when establishing a connection to submit a message. This service counters the threat of masquerade (i.e., impersonation of one MHS entity to another).

To provide *peer entity authentication*, the connection initiator transfers either simple authentication credentials (i.e., passwords) or strong authentication credentials (i.e., signatures) to the connection recipient. If strong authentication is used, the signature is applied to an asymmetric token, called a *bind* token.

In the bind token's signed-data, the connection initiator places a non-repeating number. This number allows the recipient to detect replay threats. For example, if a recipient receives a connection request where the token's non-repeating number duplicates a number received previously, the recipient can assume that the connection request is a replay of the previous connection.

The initiator may use the bind token's encrypted-data to transfer secret information, such as a symmetric encryption key. The communicating parties can use this key to encrypt data transferred across the connection (see Section 5.3.3 for an example).

The connection initiator signs the bind token using its private asymmetric key. The recipient validates the token signature using the initiator's public key certificate. This certificate may be registered with the recipient, transferred during the authentication process, or distributed by some other means.

5.2.2 Security Context

During the peer authentication process, the initiator may propose a *security context.* A security context is a set of security labels which can determine the sensitivity of messages passed over an association. If the initiator has registered a set of security labels with the connection recipient, the proposed security context must be a subset of the registered labels.

The following examples illustrate how security contexts control the transfer of messages over an association, and, in specific instances, counter the threat of message misrouting. If a security context is established between an originator and the originating MTA, the originator may only be allowed to submit messages with security labels permitted by the security context. If a security context is established between two MTA's, the transfer of messages and reports may be determined by the security label of the message or report, and the security context. This allows security labels to be used for routing purposes; only trusted MTAs (i.e., MTAs capable of establishing a security context) will be used to route a message. If a security context is established between a recipient and the delivering MTA, the MTA may only be allowed to deliver messages and reports with security labels permitted by the security context. If the security label for a message is allowed by the recipient's registered security labels, but not by the recipient's current security context, the MTA may retain the message for delivery at a later time.

5.3 Origin Authentication

Origin authentication is a set of security services allowing communicating parties to authenticate their identities. It comprises the following services: *message origin authentication, report origin authentication, proof of submission,* and *proof of delivery.*

5.3.1 Message Origin Authentication

Message origin authentication allows the identity of a message originator to be verified. This service counters the threat of masquerade (i.e., impersonation of the message originator). Since origin authentication has limited utility without content integrity, the *message origin authentication* service also provides assurance that the message content has not been modified. If a security label is present in the message, this service also enables proof of association between the security label and the message. *Message origin authentication* can be provided by one of two methods: a message origin authentication check or a content integrity check.

5.3.1.1 Message Origin Authentication Check

The message origin authentication check allows the identity of a message originator to be verified by the message recipient(s), and any MTA transferring the message. It is provided on a per-message basis using an asymmetric encryption technique.

The message origin authentication check is a digital signature included in the message envelope. The originator computes the signature as a function of the message content, the message content identifier (an optional attribute generated by the originator to facilitate the correlation of a message with any reports it may provoke), and the message security label. If the message content is encrypted, the signature is computed as a function of the encrypted content. Thus, the identity of the originator can be confirmed without the need to see the plaintext content.

If the signature is computed using the plaintext content, the message origin authentication check also provides *non-repudiation of origin* (see Section 5.6). This provision is not maintained if the signature is computed using the encrypted message content. The message originator, although unable to deny sending the encrypted content, can deny that the content decrypted by the recipient is the same as the original plaintext content.

The message origin authentication check is computed using the originator's private asymmetric encryption key. The service places no restrictions on the originator regarding which asymmetric algorithm is used. The originator conveys the object identifier for the algorithm, any input parameters required by the algorithm (e.g., an RSA message digest identifier), and the signature generated by the algorithm, in the message envelope.

The message recipient(s), and any MTA transferring the message, can validate the signature using the originator's public key certificate. This certificate may be transferred in the message envelope, or obtained by some other means.

5.3.1.2 Content Integrity Check

The content integrity check allows the identity of the originator to be verified by the message recipient(s), and possibly by any MTA transferring the message. It is provided (typically) on a per-recipient basis, using either symmetric or asymmetric encryption techniques.

The content integrity check is a cryptographic checksum included in the message token. A distinct token can be generated for each message recipient. Unlike the message origin authentication check, the content integrity check must be computed as a function of the plaintext message content. The originator places the check in the token's signed-data, or if the secrecy of the check is required, in the token's encrypted-data.

The originator may choose either a symmetric or asymmetric encryption algorithm to compute the check. If the originator chooses a symmetric encryption algorithm, a symmetric encryption key is used by the message originator to compute the check, and by the message recipient(s) to validate the check. This key can be transferred in the token's encrypted-data, or distributed by some other means (e.g., by prior agreement). Since only the originator and the recipient(s) share this key, no MTA transferring the message can authenticate the message.

If the content integrity check is computed with an asymmetric encryption algorithm (i.e., is a digital signature), the originator's private asymmetric key is used to generate the check. The recipient validates the check using the originator's public key certificate. This certificate may be transferred in the message envelope, or obtained by some other means. Providing the originator does not transfer the check in the token's encrypted-data, any MTA handling the message can validate the check.

The content integrity check can be computed using any symmetric or asymmetric algorithm understood by both the originator and the recipient. All information relevant to the algorithm can be conveyed in the message token.

By placing the content integrity check in the token's signed-data or encrypted-data, the originator provides *non-repudiation of origin* (see Section 5.6) in addition to *origin authentication.* This is because the check is computed using the plaintext content, then signed by the originator. The originator may also transfer a security label with the content integrity check in either the token's signed-data or encrypted-data, to bind the security label to the message content.

5.3.2 Report Origin Authentication

The *report origin authentication* service enables a message originator to authenticate the origin of a delivery/non-delivery report. This service counters the threat of masquerade (i.e., impersonation of the report originator). It is provided to the message originator, as well as to any MTA transferring the report, on a per-report basis using an asymmetric encryption technique. If a security label is present in the report, the service binds the security label to the report.

The reporting MTA provides this service by generating a report origin authentication check (i.e., a digital signature) and sending it in the report. The report is signed if a message origin authentication check is present in the subject message. The report signature is computed as a function of the content identifier and security label of the subject message, the name of the recipient, and for:

a delivery report: the time the message was delivered, and if requested by the originator, *proof of delivery* (see Section 5.3.4),

a non-delivery report: the reason and diagnostic for non-delivery.

The report origin authentication check is derived using the reporting MTA's private asymmetric encryption key. The check is validated by the originator of the subject message, and any MTA transferring the message, using the MTA's public key certificate. This certificate may be transferred in the report, or obtained by some other means.

5.3.3 Proof of Submission

Proof of submission allows a message originator to obtain proof that its MTA submitted a message for delivery to the intended recipient(s). This service counters the threat of masquerade (i.e., impersonation of an MTA to an MTS user). It is provided on a per-message basis using symmetric or asymmetric encryption techniques.

The message originator requests the service by submitting a *proof of submission* request with a message. The originator's MTA returns the proof in an acknowledgement. The proof is computed as a function of the submitted message arguments (i.e., the submitted message without the content), the message identifier (which is added by the MTA), and the time the message was submitted.

To generate the proof using an asymmetric encryption algorithm, the MTA signs the acknowledgement using its private asymmetric key. The message originator validates the signature using the MTA's public key certificate. This certificate may be registered with the originator, transferred in the acknowledgement, or obtained by some other means. An asymmetric *proof of submission* also provides *non-repudiation of submission* (see Section 5.6).

If the message originator transferred a symmetric encryption key to the MTA during the authentication process (see Section 5.2.1), the MTA can compute the *proof of submission* using this key. A symmetric *proof of submission* does not provide *non-repudiation of submission.*

5.3.4 Proof of Delivery

Proof of delivery allows a message originator to verify that a message was delivered to the intended recipient(s). This service counters the threat of falsely acknowledged receipt. It is provided on a per-recipient basis using symmetric or asymmetric encryption techniques.

The message originator requests the service from each message recipient (i.e., proof may be requested from some recipients, but not from others). The recipient returns the proof in a delivery report. Although the report is created by the delivering MTA, the proof included in the report is generated by the recipient MTS user (e.g., the recipient UA). The proof is computed as a function of the recipient's name, the time the message was delivered, and the following information from the subject message: the content identifier, the security label, and the plaintext content.

To generate the proof using an asymmetric encryption algorithm, the recipient signs the report using its private asymmetric key. The message originator validates the signature using the recipient's public key certificate. This certificate may be transferred in the report, or obtained by some other means. An asymmetric *proof of delivery* also provides *non-repudiation of delivery* (see Section 5.6).

If a symmetric encryption algorithm is used, the recipient computes the *proof of delivery* using a symmetric encryption key. The originator uses the same key to validate the proof. The X.400 Recommendations do not define how this key is distributed between the communicating parties. A symmetric *proof of delivery* does not provide *non-repudiation of delivery.*

A message recipient is not mandated to return *proof of delivery.* That is, even if the originator requests the service, the recipient has the option of not returning the proof. Thus, not receiving *proof of delivery* does not imply non-delivery of the subject message.

5.4 Data Integrity

Data integrity is a set of security services verifying that the content of a message has not been modified, and if a sequence of messages is transferred, that the sequence has been preserved. It comprises the *content integrity* and *message sequence integrity* services.

5.4.1 Content Integrity

The *content integrity* service allows an originator to provide proof that the content of a single message has not been modified. As mentioned previously, *content integrity* is meaningless to a user without *origin authentication. Content integrity*, on its own, authenticates the content of a message; however, the message may have been submitted by an imposter. *Origin authentication*, on its own, verifies the identity of message originator; however, the content received may not match the content originated. Thus, *content integrity* and *origin authentication* are provided by the same security mechanisms. These mechanisms, the content integrity check and the message origin authentication check, are described in Section 5.3.1.

5.4.2 Message Sequence Integrity

Message sequence integrity provides proof that the ordering of a sequence of messages sent from an originator to a recipient has been preserved. This service counters message sequencing threats, such as the replaying and re-ordering of messages. It is provided on a per-recipient basis using symmetric or asymmetric encryption techniques.

To provide the service, the message originator generates a sequence number, which identifies the position of the message in the sequence. This number is transferred in the message token's signed-data, or if the secrecy of the number is required, in the token's encrypted-data. Each originator/recipient pair using this service maintains a distinct pair of sequence numbers.

5.5 Data Confidentiality

Data confidentiality is a set of services used to protect data against unauthorized disclosure. It comprises the *content confidentiality* and *message flow confidentiality* services.

5.5.1 Content Confidentiality

Content confidentiality prevents the disclosure of the plaintext content of a message to any party other than the intended recipient(s). It is provided on a per-message basis using an asymmetric or symmetric encryption technique. The encrypted content is unintelligible to any MTA handling the message.

If the originator chooses an asymmetric algorithm, the recipient's public asymmetric key is used to encrypt the message content. The recipient uses its private asymmetric key to decrypt the content. If an asymmetric encryption algorithm is used, the message can only be addressed to a single recipient (i.e., the recipient whose private key is paired with the public key used to perform the encryption).

If the originator chooses a symmetric algorithm, delivery to multiple recipients is possible. The originator encrypts the content using a symmetric encryption key. This key may be distributed to each message recipient by placing the key in the encrypted-data of the message token for that recipient. The key may also be distributed by some other means (e.g., by prior agreement).

The message originator can encrypt the content using any symmetric or asymmetric algorithm understood by both the originator and the recipient. All information relevant to the algorithm, such as the algorithm's object identifier and any input parameters, can be conveyed in the message envelope or the signed-data of the message token.

5.5.2 Message Flow Confidentiality

Message flow confidentiality allows the message originator to conceal the flow of a message through the MHS, protecting against information that may be derived from its observation. This service counters the threats of traffic analysis and loss of anonymity of the communicating parties. It is provided by a technique called *double enveloping.*

To provide this service, the message originator specifies that the content of a message is itself a complete message (usually encrypted). The recipient on the outer envelope, upon receiving the message, forwards the message to the recipient named on the inner envelope. Double enveloping only provides a limited *message flow confidentiality* service. A more comprehensive service would include *traffic padding* and *routing control*, which are outside the scope of the X.400 Recommendations.

5.6 Non-repudiation

Non-repudiation services provide unforgeable evidence that a specific action occurred. The MHS provides the following non-repudiation services: *non-repudiation of origin*, *non-repudiation of submission*, and *non-repudiation of delivery. Non-repudiation of origin* protects against any attempt by a message originator to deny sending a message. *Non-repudiation of submission* protects against any attempt by an MTA to deny that a message was submitted for delivery. *Non-repudiation of delivery* protects against any attempt by a message recipient to deny receiving a message.

The non-repudiation services are similar to their weaker proof counterparts (i.e., *proof of submission, proof of delivery,* and *message origin authentication*); however, non-repudiation provides stronger protection, because the proof can be demonstrated to a third party. Digital signatures are used to provide non-repudiation. For example, if a recipient returns *proof of delivery* by signing a report, *non-repudiation of delivery* is also provided. Since only the recipient's private asymmetric key could have generated the signature, the signature provides unforgeable evidence of message delivery. Symmetric encryption cannot guarantee non-repudiation. Since both the originator and recipient share the symmetric encryption key, either party can generate the proof.

The exact mechanisms used to provide *non-repudiation of origin, non-repudiation of submission*, and *non-repudiation of delivery* are described in Sections 5.3.1.1, 5.3.3, and 5.3.4 respectively. Non-repudiation services may also be provided by a third party notary; however, third party notaries are outside the scope of the X.400 Recommendations.

5.7 Security Management

Throughout this chapter, references have been made to MHS *security management* services, namely, the *registration of security labels*, and the *registration of credentials.* For completeness, these services are described in this section.

The *registration of security labels* service allows an MTS user to convey a set of security labels that are maintained by its MTA. The MTA uses these labels primarily to control the delivery of messages to the MTS user.

The *registration of credentials* service allows peer MHS entities (i.e., an MTS user and its MTA) to register credentials with each other. For simple authentication, credentials comprise the password associated with the entity. For strong authentication, credentials comprise the entity's public key certificate. These credentials are used primarily when the peer entities are establishing an association.

Chapter 6

X.400 Security Limitations

The previous chapter described specific X.400 services that counter threats to the MHS. Although the services employ a broad scope of security mechanisms, there are some limitations to the 1988 X.400 security architecture. These limitations pertain to the token, the message store, and some services provided by the MTS that access the content of messages.

One security limitation is that data in the token is encrypted before it is signed. This is considered a bad practice, because the recipient can only authenticate the encrypted data, not the plaintext data. In a worst case scenario, a malicious party can intercept a message, and create a new message keeping the encrypted content of the original message, but generating a new message token. The new token would be signed by the malicious party. Under these circumstances, the message recipient could be fooled into believing that the malicious party originated the message. It should be noted that depending on how the security services are implemented, this scenario can be avoided.

A second limitation pertains to the MS (Message Store). An MS can accept the delivery of messages on behalf of a UA. If a message originator requests *proof of delivery* for a message whose content is encrypted, and the message is delivered to an MS, the MS would require access to the encryption key to provide the service (the proof must be computed using the plaintext content). This would involve providing the MS with the recipient's private asymmetric key, or the symmetric key shared between the originator and the recipient. Neither case is desirable from a security standpoint. This is one scenario where a recipient (i.e., the MS) might ignore the originator's request for *proof of delivery*.

A final limitation pertains to MTS services which access the message content. An MTA may perform conversion on incoming messages, such as converting telex data to IA5 data. Any type of conversion invalidates integrity checks. Also, if the content is encrypted, the conversion cannot be performed without the MTA first decrypting the content. To decrypt the content, the MTA would require access to the encryption key, which is not desirable from a security standpoint.

Bibliography

[1] CCITT 1988 X.400, Message handling system and service overview.

[2] CCITT 1988 X.402, Message handling systems: Overall architecture.

[3] CCITT 1988 X.411, Message handling systems - Message transfer system: Abstract service definition and procedures.

[4] CCITT 1988 X.509, The Directory - Authentication Framework.

[5] Ankney, Richard. Security Services in Message Handling Environments. The Messaging Technology Report, Vol. 1, No. 5, June 1992.

[6] Mitchell, Christopher, Walker, Michael, Rush, David. CCITT/ISO Standards for Secure Message Handling. IEEE Journal on Selected Areas in Communications, Vol. 7 No. 4, May, 1989.