This document is not an ASTM standard and is intended only to provide the user of an ASTM standard an indication of what changes have been made to the previous version. Because it may not be technically possible to adequately depict all changes accurately, ASTM recommends that users consult prior editions as appropriate. In all cases only the current version of the standard as published by ASTM is to be considered the official document.

Designation: E 2085 – 00a

An American National Standard

Standard Guide on
Security Framework for Healthcare Information[1]

This standard is issued under the fixed designation E 2085; the number immediately following the designation indicates the year of original adoption or, in the case of revision, the year of last revision. A number in parentheses indicates the year of last reapproval. A superscript epsilon (ϵ) indicates an editorial change since the last revision or reapproval.

1. Scope

1.1 This guide covers a framework for the protection of healthcare information. It addresses both storage and transmission of information. It describes existing standards used for information security which can be used in many cases, and describes which (healthcare–specific) standards are needed to complete the framework. Appropriate background information on security (and particularly cryptography) is included. The framework is designed to accommodate a *very large* (national or international), *distributed* user base, spread across many organizations, and it therefore recommends the use of certain (scaleable) technologies over others.

1.2 Electronic information exchange and sharing of data in has been the backbone of industries such as financial institutions for several years. Cost cutting measures and a real need for sharing of information are driving healthcare services toward increased use of computer-based information systems. One of the requirements for the ability to share and exchange healthcare information is that the information be protected.

1.3 Selection of standards was performed using the following criteria, which are described in more detail in 4.2.

1.3.1 Security requirements are defined in this framework, and (in some cases) in additional ASTM guidelines.

1.3.2 ASTM standard specifications are used to define protocols and message formats in support of interoperability.

1.3.3 Existing standards will be reused or extended whenever possible.

1.3.4 This framework does not address policy issues. ASTM Subcommittee E31.17 is writing standards that address these issues.

2. Referenced Documents

2.1 *ASTM Standards:*

[1] This guide is under the jurisdiction of ASTM Committee E31 on Healthcare Informatics and is the direct responsibility of Subcommittee E31.20 on Data and System Security for Health Information.

Current edition approved ~~April~~ Oct. 10, 2000. Published ~~June~~ November 2000. Originally published as E 2085–00. Last previous edition E 2085–00.

E 1238 Specification for Transferring Clinical Observations Between Independent Computer Systems[2]

E 1384 Guide for Content and Structure of the Computer-Based Patient Record[2]

E 1762 Guide for Electronic Authentication of Healthcare Information[2]

E 1985 Guide for User Authentication and Authorization[2]

E 1986 Guide for Information Access Privileges to Health Information[2]

E 2084 Specification for Authentication of Healthcare Information Using Digital Signatures[2]

E 2086 Guide for Internet and Intranet Healthcare Security[2]

2.2 *IETF Standards:*[3]

RFC 1510 Kerberos Authentication Service

RFC 1777 Lightweight Directory Access Protocol (v2)

RFC 2251 Lightweight Directory Access Protocol (v3)

RFCs 1901–1910 Simple Network Management Protocol

RFC 1945 Hypertext Transfer Protocol

RFC 1964 Kerberos v5 GSS-API Mechanism

RFC 2025 GSS–API Simple Public Key Mechanism (SPKM)

RFC 2078 Generic Security Services Application Program Interface

RFC 2246 The TLS Protocol Version 1.0

RFC 2401 Security Architecture for the Internet Protocol

RFC 2402 IP Authentication Header

RFC 2403 The Use of HMAC-MD5–96 within ESP and AH

RFC 2404 The Use of HMAC-SHA-196 within ESP and AH

RFC 2406 IP Encapsulating Security Payload (ESP)

RFC 2407 The Internet IP Security Domain of Interpretation for ISAKMP

RFC 2408 Internet Security Association and Key Management Protocol (ISAKMP)

RFC 2409 The Internet Key Exchange (IKE)

RFC 2440 OpenPGP Message Format

RFC 2451 The ESP CBC-Mode Cipher Algorithms

RFC 2527 Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework

RFC 2259 Internet X.509 Public Key Infrastructure Operational Protocols—LDAPv2

RFC 2560 Internet X.509 Public Key Infrastructure Online Certificate Status Protocol

RFC 2630 Cryptographic Message Syntax

RFC 2631 Diffie-Hellman Key Agreement Method

RFC 2632 S/MIME Version 3 Certificate Handling

RFC 2633 S/MIME Version 3 Message Specification

RFC 2634 Enhanced Security Services for S/MIME

2.3 *ISO Standards:*[4]

ISO 8824–1 Specification of Abstract Syntax Notions One (ASN.1)

ISO 8825–1 Specification of Basic Encoding Rules for Abstract Syntax Notions One (ASN.1)

ISO/IEC 7498–2 Security Architecture

ISO/IEC 8879 Standard Generalized Markup Language (SGML)

ISO/IEC 9735 Electronic Data Interchange for Administration, Commerce and Transport (EDIFACT)–Application Level Syntax Rules (Parts 5–10)

ISO/IEC 9595 Information Technology–Open Systems Interconnection–Common Management Information Service Definition

ISO/IEC 9596 Information Technology–Open Systems Interconnection–Common Management Information Protocol Specification

ISO/IEC 10164–7 Information Technology–Open Systems Interconnection–Systems Management: Security Alarm Reporting Function

ISO/IEC 10164–8 Information Technology–Open Systems Interconnection–Systems Management: Security Audit Trail Function

ISO/IEC 11586 Generic Upper Layers Security (4 parts)

ISO/IEC 11577 Network Layer Security Protocol

ISO/IEC 10736 Transport Layer Security Protocol

ITU–T X.509 Directory Authentication

[2] *Annual Book of ASTM Standards*, Vol 14.01.

[3] Available online at ftp: //ds.internic.net.

[4] Available from ISO, 1 Rue de Varembe, Case Postale 56, CH 1211, Geneve, Switzerland.

2.4 *ANSI Standards:*[5]

X3.92 Data Encryption Standard
X9.30 Part 1 Public Key Cryptography Using Irreversible Algorithms: Digital Signature Algorithm
X9.30 Part 2 Public Key Cryptography Using Irreversible Algorithms: Secure Hash Algorithm (SHA–1)
X9.31 Reversible Digital Signature Algorithms
X9.42 Management of Symmetric Keys Using Diffie–Hellman
X9.44 Key Establishment Using Factoring-Based Public Key Cryptography for the Financial Services Industry
X9.57 Certificate Management
X9.55 Extensions to Public Key Certificates and CRLs
X9.52 Triple DES Modes of Operation
X9.62 Elliptic Curve Digital Signature Algorithm
X12 Electronic Data Interchange
X12.58 Security Structures (version 2)
X.25 Interface between Data Terminal Equipment (DTE) and Data Circuit-Terminating Equipment (DCE) Operating in the Packet Mode and Connected to Public Networks by Dedicated Circuits
X.500 Open Systems Interconnection: The Directory

2.5 *Other Standards and Publicly Available Specifications:*[6]

FIPS PUB 46–3 Data Encryption Standard
FIPS PUB 74 Guidelines for Implementing and Using the NBS Data Encryption Standard
FIPS PUB 81 DES Modes of Operation
FIPS 140–1 Security Requirements for Cryptographic Modules
FIPS PUB 180–1 Secure Hash Algorithm
FIPS PUB 186 Digital Signature Standard
IEEE 802.10 *Interoperable LAN/MAN Security (SILS)*, 1992–1996 (multiple parts)
NIST MISPC Minimum Interoperability Specification for PKI Components Version 1

3. Terminology

3.1 *Definitions:*

3.1.1 *algorithm*—a clearly specified mathematical process for computation; a set of rules which, if followed, will give a prescribed result.

3.1.2 *asymmetric cryptography*—cryptographic algorithm that uses two related keys, a public key and a private key; the two algorithm keys have the property that, given the public key, it is computationally infeasible to derive the private key.

3.1.3 *authentication*—the corroboration that the source of data received is as claimed.

3.1.4 *authorization*—the granting of rights.

3.1.5 *cipher text*—data in its enciphered form.

3.1.6 *clear text*—data in its original, unencrypted form.

3.1.7 *confidentiality*—the property that information is not made available to or disclosed to unauthorized individuals, entities, and processes.

3.1.8 *cryptography*—the discipline which embodies principles, means, and methods for the transformation of data in order to hide its information content, prevent its undetected modification, prevent its unauthorized use, or a combination thereof.

3.1.9 *data integrity*—a property whereby data has not been altered or destroyed.

3.1.10 *decryption*—a process of transforming ciphertext (unreadable) into plain text (readable).

3.1.11 *digital signature*—a cryptographic transformation of data which, when associated with a data unit, provides the services of origin authentication, data integrity, and signer non–repudiation.

3.1.12 *encryption*—a process of transforming plain text (readable) into cipher text (unreadable) for the purpose of security or privacy.

3.1.13 *encryption key*—a binary number used to transform plain text into cipher text.

3.1.14 *gateway*—a computer system or other device that acts as a translator between two systems that do not use the same communications protocols, data formatting, structures, languages, or architecture, or a combination thereof.

3.1.15 *non–repudiation*—this service provides proof of the integrity and origin of data (both in an unforgeable relationship) which can be verified by any party.

3.1.16 *plain text*—data in its original, unencrypted form.

3.1.17 *repudiation*—the denial by a user of having participated in part or all of a communication (see *non–repudiation* , which has the opposite meaning).

3.1.18 *replay*—the process of sending a previously sent message as a method of perpetrating a fraud.

[5] Available from American National Standards Institute, 11 W. 42 nd St., 13th Floor, New York, NY 10036.

[6] National Technical Information Service, U.S. Department of Commerce, Springfield, VA. http://csrc.nist.gov or www.ntis.gov.

3.1.19 *security association*—the relationship between two entities which allows the protection of information communicated between the entities.

3.1.19.1 *Discussion*—This relationship includes a shared symmetric key and security attributes describing the relationship. The security association is used to negotiate the characteristics of these protection mechanisms, but does not include the protection mechanisms themselves.

3.1.20 *session*—a logical relationship between two network endpoints that supports a user or network application.

3.1.21 *subnetwork*—a network segment usually with its own address.

3.1.22 *symmetric encryption*—encryption using a single key to encrypt and decrypt which both the sender and receiver hold privately.

3.1.23 *virtual private network*—a network using public data network or the Internet as a carrier that acts as if a dedicated point to point network.

3.1.23.1 *Discussion*—Cryptography is normally used to protect data.

3.2 *Acronyms:Acronyms:*

3.2.1 *AH*—Authentication Header

3.2.2 *API*—Application Programming Interface

3.2.3 *ASTM*—American Society for Testing and Materials

3.2.4 *ATM*—Asynchronous Transfer Mode

3.2.5 *CA*—Certificate Authority

3.2.6 *CMIP*—Common Management Information Protocol

3.2.7 *CMS*—Cryptographic Message Syntax

3.2.8 *CORBA*—Common Object Request Broker Architecture

3.2.9 *DSA*—Digital Signature Algorithm

3.2.10 *DES*—Data Encryption Standard

3.2.11 *EDI*—Electronic Data Interchange

3.2.12 *ESP*—Encapulating Security Payload

3.2.13 *FTP*—File Transfer Protocol

3.2.14 *GSS*—Generic Security Services

3.2.15 *HMAC*—Hashed Message Authentication Code

3.2.16 *HTTP*—HyperText Transfer Protocol

3.2.17 *IDUP*—Independent Data Unit Protection

3.2.18 *IETF*—Internet Engineering Task Force

3.2.19 *IP*—Internet Protocol

3.2.20 *IPS*—Internet Protocol Suite

3.2.21 *IPSEC*—Internet Protocol Security

3.2.22 *KRA*—Key Release Agent

3.2.23 *LAN*—Local Area Network

3.2.24 *LDAP*—Lightweight Directory Access Protocol

3.2.25 *MD*—Message Digest

3.2.26 *MIME*—Multipurpose Internet Mail Extension

3.2.27 *MSP*—Message Security Protocol

3.2.28 *NLSP*—Network Layer Security Protocol

3.2.29 *OSI*—Open Systems Interconnection

3.2.30 *PCT*—Private Communications Technology

3.2.31 *PIN*—Personal Identification Number

3.2.32 *PKI*—Public Key Infrastructure

3.2.33 *PRNG*—Pseudo Random Noise Generator

3.2.34 *RFC*—Requests for Comment

3.2.35 *RSA*—Rivest, Shamir, and Adelman

3.2.36 *SHA-1*—Secure Hash Algorithm

3.2.37 *S–HTTP*—Secure HyperText Transfer Protocol

3.2.38 *S/MIME*—Secure/Multipurpose Internet Mail Extension

3.2.39 *SMTP*—Simple Mail Transfer Protocol

3.2.40 *SSL*—Secure Socket Layer

3.2.41 *TCP*—Transmission Control Protocol

3.2.42 *TLSP*—Transport Layer Security Protocol

3.2.43 *VPN*—Virtual Private Network

3.2.44 *WAN*—Wide Area Network

3.2.45 *WWW*—World Wide Web

4. Significance and Use

4.1 This guide presents a framework for securing healthcare information of all kinds. Specific existing standards are identified which accommodate many cases, and requirements for new standards are identified. An organization's security policy will determine when these standards are to be used, based on risk analysis.

4.2 Many standards have been defined by other standards bodies such as ISO, ITU, and the IETF. There are also a variety of de facto standards and publicly available specifications such as the PKCS documents from RSA Laboratories.[7] This framework recommends appropriate existing standards where possible, using the following criteria:

4.2.1 High level requirements for security are defined in this framework. In some cases, guidelines defining additional requirements will be needed. Guide E 1762 is an example of such a guideline for authentication of healthcare information.

4.2.2 Formal standards (for example, ASTM "standard specifications") are only required where information is exchanged between systems, to ensure interoperability. These standards define protocols and message formats.

4.2.3 If there are no healthcare specific requirements for some security service, one or more existing standards will be recommended, as is.

4.2.4 Where existing healthcare standards (for example, HL7) use specific underlying protocols and technologies, security mechanisms already defined for those protocols will be identified and recommended.

4.2.5 Healthcare specific requirements will be met, if possible, by extending existing standards. Specification E 2084 is an example of this approach.

4.2.6 Preference is given to standards which have the greatest market acceptance and maturity.

4.2.7 Standards which involve the use of cryptography shall be, to the extent possible, algorithm–independent. This can be accomplished by, for example, signaling the algorithms used within the protocol or message format.

4.2.8 The total number of security standards needed will be minimized, subject to the previous requirements.

4.2.9 Policy issues are not addressed, although these technical standards shall accommodate any potential variations in policy allowed by other standards. Policy may be the subject of security standards produced by other groups, such as ASTM Subcommittee E31.17.

4.3 This guide assumes the standard distributed environment, including multiple heterogeneous systems, interconnected by a network. Regardless of the network protocols used, it is useful to separate functionality into the following three components:

4.3.1 *Semantics*~~:~~—This includes the application data and behavior model. At this level, security is viewed as a pervasive service provided by the application's infrastructure. An application's security policy would define access rules for the data, as well as constraints on its behavior. These would be implemented using security mechanisms provided by the infrastructure, such as access control lists and secure communications protocols.

4.3.2 *Syntax*~~:~~—This includes rules for encoding data for transport between systems (for example, ASN.1 basic encoding rules (ISO/IEC 8824 and 8825), HL7 message and field formats). Security mechanisms generally require some additional syntax. In many cases, an entire message or document can be encapsulated in a security envelope, leaving the original structure intact inside the envelope. While standardized encoding rules are also required for performing some cryptographic operations (such as digital signature), applications generally are free to use any syntax internally.

4.3.3 *Transport*~~:~~—This includes movement of data (encoded using some syntax) between systems. This typically involves adding more data elements related to the communications, for example, message headers and session identifiers.

4.4 This document is divided into several parts. Section 5 presents a security overview including threats and security services. Section 6 presents Communication Security. Local Security is presented in Section 7.

5. Security Overview

5.1 This section presents an overview of the threats addressed by a security architecture, as well as the services and mechanisms used to counter these threats. Many of these threats attack information in transit between systems (particularly those connected using open networks), and we use the generic term *message* to refer to any such data.[8] A description of the security services and mechanisms used to counter various threats and the placement of these security services in the OSI model is provided in the OSI Security Architecture (ISO/IEC 7498–2).

5.2 The following subsections discuss threats to a system, and appropriate security services to counter these threats. Detailed discussions of two particularly important security tools (access control mechanisms and cryptography) are also included.

5.3 *Threats*—This section describes the principal threats to a system. In some cases, security services can prevent an attack; in other cases, they merely detect an attack.

5.3.1 *Masquerade* occurs when an entity successfully pretends to be another entity. This includes impersonation of users or system components, as well as falsely claiming origination or acknowledging receipt of a message or transaction. For example, an adversary might masquerade as a hospital employee to gain access to medical records. Masquerade, then, facilitates the following described attacks:

5.3.2 *Modification of Information* can include modification of message or data content, as well as destruction of messages, data, or management information. The adversary in 5.3.1 could potentially modify medical records.

[7] Available from RSA Data Security, 100 Marine Parkway, Redwood City, CA 94065. http://www.rsa.com.

[8] Ford, Warwick, *Computer Communications Security: Principles, Standard Protocols and Techniques*, Prentice Hall, 1994.

5.3.3 *Message Sequencing* threats occur when the order of messages is altered. Such threats include replay, pre–play, and delay of messages, as well as reordering of messages. The adversary might capture a password message when a legitimate user logs on, and later replay it to masquerade as that user.

5.3.4 *Unauthorized Disclosure* threats include revealing message contents or other data, as well as information derived from observing traffic flow, as well as revealing information held in storage on an open system. While masquerading as a legitimate user, the adversary can access information for which he is not authorized.

5.3.5 *Repudiation* occurs when a user or the system denies having performed some action, such as origination or reception of a message. For example, a user might deny having modified a portion of the medical record.

5.3.6 *Denial of Service* threats prevent the system from performing its functions. This may be accomplished by attacks on the underlying communications infrastructure, attacks on the underlying applications, or by flooding the system with extra traffic.

5.4 *Security Services*—The following services protect against the threats described in 5.3.1-5.3.6:

5.4.1 *Peer Entity Authentication* provides proof of the identity of communicating parties. On a single system, users are authenticated during logon. For distributed environments, various types of authentication exchanges have been discussed in the literature; most are based on digital signatures or other cryptographic mechanisms.

5.4.2 *Data Origin Authentication* counters the threat of masquerade, and is provided using digital signatures or other cryptographic integrity mechanisms.

5.4.3 *Access Control* counters the threat of unauthorized disclosure or modification of data. This is particularly appropriate on an end system. A variety of access control strategies can be found in Guide E 2086, Ford,[8] and Menezes et al.[9]

5.4.4 *Confidentiality* counters the threat of unauthorized disclosure, particularly during the transfer of information. Confidentiality can be applied to entire messages or to selected fields. Encryption may be used to provide this service. Note that selective field confidentiality generally requires modification of existing message structures, in contrast to encapsulation of an entire message in a secret message "envelope." For example, adding security features to ANSI X12 EDI interchanges required extensions to the existing syntax to accommodate security elements at the transaction set and functional group levels.

5.4.5 *Integrity* counters the threat of unauthorized modification of data. This can be provided with various types of integrity check values. To protect against deliberate modification, a cryptographic check value or digital signature should be used. This also provides the service of data origin authentication. As with confidentiality, this service may be applied to entire messages or selected fields. One particularly useful application of selective field integrity is message sequence integrity, in which the integrity service is applied to a sequence number or other sequencing information.

5.4.6 *Non-repudiation* of origin and delivery protect against an originator or recipient falsely denying originating or receiving a message. This service provides proof (to a third party) of origin or receipt, and is provided using digital signatures. See Table 1.

5.5 *Access Control Mechanisms*:

5.5.1 Access control mechanisms perform the following functions:

5.5.1.1 *Decide* whether a given *initiator* (such as a user) can perform some *action* (such as read) on a given *target* (such as a file).

5.5.1.2 *Enforce* this access control decision.

5.5.2 In general, an access control decision can make use of information associated with the initiator (for example, the user's ID), information associated with the target (for example, the file name), the type of action requested, and other information associated with the request (for example, time of day). As a simple example, many operating systems allow an access control list to be associated with a file or directory; the list defines which users can perform which actions on the file. As another example, many military systems associate a classification with each target (for example, confidential, secret, top secret) and a clearance with each initiator. The target can be accessed only if the initiator's clearance is at least equal to the target's classification.

5.5.3 Depending on the application, it may be desirable to group initiators together by role or organization. This can greatly simplify administration of access control information, for example, by using a role name in a single access control list entry rather

[9] Menezes, Alfred, van Oorschot, Paul C., Vanstone, Scott A., *Handbook of Applied Cryptography*, CRC Press, 1997.

TABLE 1 Security Threats vs. Services

NOTE 1—The data secured by the integrity service shall include sequence numbers or other sequencing information.

Threat	Security Service
Masquerade	Data Origin Authentication, Peer Entity Authentication
Modification of Information	Integrity
Message Sequencing	Integrity (see Note 1)
Unauthorized Disclosure	Confidentiality
Repudiation	Non-Repudiation
Denial of Service	Not addressed in this provisional guide

than a separate entry for each user with that role. Similarly, granularity of access to the target might vary, from an entire database or directory, to specific files, specific records within files, or even specific fields within a record.

5.5.4 On a single system, access control is typically enforced by the operating system. As an extra level of protection, one could also encrypt sensitive data (see 5.5.5) so that only users with the appropriate key could decrypt and access it. This would protect against attackers who subverted the operating system access controls.

5.5.5 In the distributed environment, it is still entirely feasible to attach an access control list to a target, but the list must identify the user relative to the entire system (for example, "user X on system Y"). Other approaches are also possible. For example, while the access control enforcement function would still be performed on the system where the target resides, the decision could be made on the initiator's system. The initiator's system might then issue appropriate "credentials" indicating which targets the initiator can access. This "capability" model minimizes the complexity on the target's system (which simply checks credentials rather than needing to maintain access control lists), at the expense of more complexity on the initiator's system. Taking the distributed scenario a bit farther, Ford and Weiner[10] describe a system where access control information (of any type) is bound to an object and travels with it. This is discussed in more detail in 6.2.4.3.

5.6 *Cryptography*:

5.6.1 Many security services are provided using cryptography. Cryptography scrambles and unscrambles data using *keys*. The amount of effort to unscramble data without having the correct key is proportional to the length of the key. Thus, cryptographic algorithms should use keys of sufficient length to preclude such a "brute–force" attack.[9]

5.6.2 In *symmetric* (conventional) cryptography, the sender and recipient share a secret key. This key is used by the originator to encrypt a message and by the recipient to decrypt a message. DES is an example of a symmetric cryptosystem. The shared key shall somehow be conveyed between the two parties. Mechanisms to do this include the following:

5.6.2.1 *Key Transport*— encrypting the key under an existing key.

5.6.2.2 *Key Agreement*— see 5.6.5.5.

5.6.2.3 *Manual Distribution*—for example, at initial installation.

5.6.3 In *asymmetric* (public key) cryptography, different keys are used to encrypt and decrypt a message. Each user is associated with a pair of keys. To provide confidentiality, one key (the *public key*) is publicly known and is used to encrypt messages destined for that user, and one key (the *private key*) is known only to the user and is used to decrypt incoming messages. While there is no need to distribute private keys, since each entity can generate its own, there is a need to distribute public keys in such a way that users can be sure to whom the keys belong (see 5.6.6).

5.6.4 Authentication can be provided using a public key system, using the concept of *digital signatures* described in 5.6.5.1. RSA is the most well known asymmetric algorithm. Since the public key need not (indeed cannot) be kept secret, it is no longer necessary to secretly convey a shared encryption key between communicating parties prior to exchanging confidential traffic or authenticating messages.

5.6.5 The following security mechanisms are constructed from symmetric and asymmetric cryptosystems:

5.6.5.1 A *digital signature* on a message is computed by hashing the message and encrypting the hash using the originator's private key. The signature can be verified using the originator's public key.

5.6.5.2 A *digital envelope* consists of a symmetric key (used for bulk encryption of a message), and, optionally, other information, encrypted under the public key of a recipient. This is an example of key transport.

5.6.5.3 *Bulk encryption* uses a symmetric algorithm to encrypt a message. Typically, a new encryption key is generated randomly for each message and conveyed to the recipient in a digital envelope.

5.6.5.4 A *message authentication code* (MAC) is a cryptographic checksum computed over a message, using a shared secret key. The MAC might be used to encrypt the message using a chaining mode of operation (where the MAC is then some portion of the last encrypted block), or the key might be used to encrypt a hash of the message.

5.6.5.5 *Key agreement* is used to compute a shared key without conveying any portion of it (even in a digital envelope) between sender and recipient. This is another type of public key algorithm, which typically uses public and private keys from both originator and recipient to generate the shared key.

5.6.5.6 For a user to identify another user by his possession of a private key, or to encrypt data using another user's public key, he must obtain the other user's public key from a source he trusts. A framework for the use of *public key certificates* was defined in ITU—T X.509. These certificates bind a user's name to a public key, and are signed by a trusted issuer called a *Certification Authority* (CA). Besides the user's name and public key, the certificate contains the issuing CA's name, a serial number, and a validity period.

5.6.6 A particularly useful public key infrastructure (PKI) would arrange CAs into a small number of hierarchies, where each CA may certify subordinate CAs as well as end users. Ideally, a user should be able to build a path of certificates from one trusted public key (for example, her CA or a "root" of a CA hierarchy) to any other user's certificate, anywhere in the world.

5.6.7 In smaller environments, such as closed systems involving a fairly small number of trading partners, a hierarchy of CAs may not be necessary. Indeed, it may be feasible for all users to "manually" exchange public keys. This "web of trust" approach is used in PGP (Menezes et al[9]).

[10] Ford, W. and Weiner, M., "A Key Distribution Method for Object-Based Protection," *2nd ACM Conference on Computer Communications and Security*, 1994.

5.6.8 Appropriate standards for algorithms, certificates, and key management mechanisms are discussed in Section 8.

6. Communications Security

6.1 In a distributed environment, there are multiple systems communicating over a network. It is not necessarily the case that a system will trust another system without, at a minimum, authenticating its identify (peer entity authentication[8,9]). Within a network, entities communicate using protocols. Frequently, these protocols are layered in order to isolate details of one layer from another. For example, media dependent protocol details are placed at the lowest layers, so that higher layers see a reliable, sequenced transport service. These higher layers, in turn, might provide dialog control and synchronization, transfer encoding and decoding, and similar functions which need to be isolated from the application. Two popular layered protocol stacks are TCP/IP and OSI. While different stacks have different numbers of layers, from a security perspective we can isolate functionality into four layers (each of which may encompass more than one layer in a real protocol stack).

6.2 *Application Level Security* :

6.2.1 Security may be placed at the application level (for example, within specific applications). It shall be placed at this level if the following situations exist:

6.2.1.1 The security services are application–specific, or

6.2.1.2 The services traverse application relays.

6.2.2 An example of 6.2.1.1 is secure file transfer applications, which deal with access control information attached to files. Another example is applications that selectively protect fields, for example, an application which encrypts only sensitive information such as patient identifiers. The major example of 6.2.1.2 is store–and–forward electronic mail, in which sender and recipient(s) never directly communicate, and in which only the content portion of a message is protected. Messages are relayed from sender to recipient via application programs called mail transfer agents or mail relays. Electronic Data Interchange (EDI) systems are also examples of this type of application.

6.2.3 Session–oriented applications are characterized by two entities establishing a connection and exchanging information in real time. When communications are complete, the connection is closed. Many peer–to–peer and client/server applications fall in this category. These applications generally expect a reliable, sequenced network transport service to be available. Several existing protocols follow that can be used for these applications:

6.2.3.1 *Simple Public Key Mechanism (SPKM) (RFC 2025)* is designed for use with any session–oriented application. It provides confidentiality, integrity, authentication (both entity and origin), and (optional) non–repudiation. This handles all peer–to–peer and client–server applications quite well. It is designed for use with the Generic Security Services API (GSS–API) (RFC 2078) discussed in 6.2.3.2. It is also recommended for use in CORBA applications, which makes it particularly appropriate for CORBA–based HL7 applications.

6.2.3.2 *Transport Layer Security (TLS) (RFC 2246)* is designed for use with client/server applications, particularly World Wide Web (WWW) applications. It provides confidentiality, integrity, and peer entity authentication, as well as key management mechanisms. It is based on the Secure Sockets Layer (SSL) protocol developed by Netscape. SSL is widely deployed (as part of most Web browsers) and so it can be used immediately to secure Web–based applications.

6.2.3.3 *Secure HTTP (S–HTTP) (Draft Secure HTTP)* defines a request/response protocol on top of the HTTP protocol (RFC 1945) used in the WWW. This protocol can secure each request/response pair separately, and provides data origin authentication, integrity, and confidentiality. It also provides non–repudiation of responses. It is based on the CMS format discussed in 6.2.4.2, and is effectively CMS with HTTP “transport syntax” preceding it. S–HTTP emphasizes record or document level protection rather than session-level protection. The S-HTTP protocol is currently a work in process in the IETF. The current Internet draft documents have expired.

6.2.3.4 For OSI networks, Generic Upper Layers Security (GULS) (ISO/IEC 11586) defines mechanisms for application layer protection of any desired type.

6.2.4 Store–and–forward applications are characterized by unidirectional traffic from sender to recipient. The sender need not establish a connection (E–mail is an obvious example), and each message is protected independently. Recommended existing standards include CMS (the format underlying a number of other standards, defined in RFC 2630 and RFC 2631), and (for certain applications) X12 and EDIFACT (ISO 9735) security. EDI security would only be used when different transaction sets or functional groups in an interchange need different protection. For example, some transaction sets might be encrypted, while others are not. This is an example of selective field protection at a fairly coarse level (ANSI X12.58).

6.2.4.1 CMS supports encryption and signature of arbitrary data. This includes support for multiple signatures and other requirements from Guide E 1762. While it is entirely usable now, term enhancements in the near future will provide even more useful functionality.

6.2.4.2 CMS is used as the basis for the S/MIME secure E–mail standard, S–HTTP (see 6.2.4), the Secure Electronic Transaction (SET) credit card transaction standard,[11] the ANSI X9.45 authorization certificate standard, and the ASTM digital

[11] This guide is under the jurisdiction of ASTM Committee E31 on Healthcare Informatics and is the direct responsibility of Subcommittee E31.20 on Data and System Security for Health Information.

Current edition approved Oct. 10, 2000. Published November 2000. Originally published as E 2085–00. Last previous edition E 2085–00.

signature standard. There is also ongoing work to migrate the DoD Message Security Protocol (MSP) to PKCS #7. S/MIME is basically CMS with MIME (E–mail) headers, as defined in RFC 2632, RFC 2633, and RFC 2634. While the current specification requires support for proprietary encryption algorithms, this problem should be fixed during IETF standardization. Alternatively, new MIME headers of more relevance to healthcare (such as HL7 message types) could be defined.

6.2.4.3 For many store–and–forward applications, there is a requirement to ensure that a received transaction or document is "authorized," that is, acceptable based on the rules and limits imposed by the application. This is easily accomplished in a centralized environment. However, in a distributed environment, it is more cost effective to convey authorization information in certificates. Guide E 1985 discusses healthcare requirements in this area. For example, it may be a requirement that either a primary–care physician, or an intern and a reviewing physician, sign off on a document prior to placing it in the official medical record. ASTM is developing a standard for this, using the data model of Specification E 1238 and Guide E 1384.

6.2.4.4 For closed systems with a small number of trading partners, PGP/MIME (RFC 2440) may be used for secure messaging. While the key management used in PGP does not scale as well as the X.509 CA hierarchy used in S/MIME, it is entirely suitable for small applications.

6.2.4.5 Paragraph 6.2.4.3 discusses a mechanism to carry access control information along with a document. The document is encrypted under a bulk encryption key. The bulk key and access control information are encrypted under the public key of a key release agent (KRA). To access the document a recipient provides any required privileges (for example, a certificate containing her name for an access control list model) to the KRA. If these privileges are satisfactory the KRA returns the bulk key to the recipient, who can then decrypt the document. As a very long term goal, defining one or more appropriate access control structures for use with the KRA model could accommodate differences in confidentiality policy among organizations (or countries). Such a structure would likely require support for selective access to portions of the document.

6.2.4.6 The selective field protection provided by X12.58 was discussed in 6.2.4.5. Another format where selective field protection would be useful is Standard Generalized Markup Language (SGML) (ISO/IEC 8879). SGML and its WWW subset, XML, allow text documents to be structured using tag fields. This ability to create semi–structured documents, as opposed to completely structured database records or completely free form text documents, is obviously very useful in the medical records area. There is ongoing discussion on use of XML in both ASTM and HL7. There is currently a joint IETF/W3C working group defining mechanisms for digitally signing XML, but no standards have yet been produced by this group.

6.3 *End–System Level Communications Security*:

6.3.1 Security may be provided at the end–system level. This would be advisable in the following situations:

6.3.1.1 The end system is trusted, but the underlying network is not trusted.

6.3.1.2 Protection is required (by security policy) for all (or most) traffic.

6.3.2 In the cases discussed in 6.3.1.1 and 6.3.1.2, end–system level security is preferable to application level security for the following reasons:

6.3.2.1 The security services are transparent to applications (no code changes).

6.3.2.2 Performance of bulk data protection services is improved, as they can operate on larger data units and handle all applications the same way.

6.3.2.3 Administration is simplified, as only a single administrator is required.

6.3.2.4 Upper layer protocol headers are protected.

6.3.3 There is a complete set of IP security standards (IPSEC) available. IPSEC is defined in RFC 2401, RFC 2402, RFC 2406, RFC 2407, RFC 2408, RFC 2409, and RFC 2451. In addition, for OSI networks, transport (TLSP) and network (NLSP) layer security protocols have been defined. TLSP is defined in ISO/IEC 10736, and NLSP is defined in ISO/IEC 11577. Note that the network layer protocol is application–independent, so these standards can be used as is. ASTM has developed a guideline recommending specific options within the (fairly complex) IPSEC protocol suite. All of these protocols provide authentication, integrity, confidentiality, and associated key management functionality. These protocols are independent of the application; no healthcare–specific requirements are foreseen at this layer.

6.4 *Subnetwork Level Security*:

6.4.1 This level of security protects data across one or more specific subnetworks. For example, one might have an environment where traffic traverses the originator's LAN, an WAN, and the recipient's LAN. Each of these could be protected individually. Reasons for using this approach include the following:

6.4.1.1 Subnetworks close to end–systems are typically trusted as much as the end–system (frequently the end–systems and subnetwork might share a security administrator). However, intervening subnetworks such as the WAN in the example in 6.4.1 are less trusted, and

6.4.1.2 This solution is generally cheapest in terms of equipment, since there are many more end–systems than there are subnetwork gateways (for example, routers).

6.4.2 IP, when used on a subnetwork basis, can make use of the IPSEC standards (see 6.3.3). Similarly, NLSP can be used in OSI environments. Proprietary solutions also exist for specific subnetwork protocols such as X.25.

6.5 *Direct Link Level Security* :

6.5.1 Direct link level security operates at the physical or (for LANs) data link layer. It would be used where there are a few untrusted links in an otherwise trusted network. While there are many products on the market, operational and equipment costs are high, since devices must be independently managed on a link–by–link basis. However, such protection is transparent to all higher level protocols.

6.5.2 Standards are available for link encryption, including standards for use of DES over asynchronous lines and frame–level LAN security (SILS), as specified in IEEE 802.10.

6.5.3 Where dedicated lines are used, physical protection of the circuits may be an alternative way to protect a link. In these situations, there would be no need for cryptographic security mechanisms.

6.6 *Placement of Security Services* :

6.6.1 Following are several properties to consider when determining the proper placement of security services, as discussed in Ford[8]:

6.6.1.1 Since upper layer traffic is typically multiplexed onto lower layer connections, it is likely that security services at lower levels will be protecting a data stream containing traffic to or from different sources and destinations. If the security policy dictates that all (or most) traffic requires a certain degree of protection, use of lower level security services is desirable for efficiency. If security is at the discretion of individual users, lower level services may not be desirable due to the cost of unnecessarily protecting data which does not require protection. In such a case, application level security is a better choice.

6.6.1.2 At lower levels, there is more knowledge of the security characteristics of particular routes and links. If these characteristics vary greatly within different portions of the network, then placing security at a lower (for example, subnetwork) level is desirable, since appropriate security services can be selected on a per-subnetwork (or per–link) basis rather than being implemented in all end systems. Use of subnetwork level security would allow gradual migration of security into existing networks.

6.6.1.3 As mentioned in 6.6.1.2, the minimum number of protection points is at the subnetwork layer. This level of security might be the most cost effective, compared to direct link level security. Placing services at the direct link layer requires security devices at the ends of every link. Placing services at higher layers requires their implementation in every end–system or sensitive application. Since much of this could be done in (relatively inexpensive) software rather than in hardware, a cost analysis should be performed to determine which approach is cheapest.

6.6.1.4 When security services are provided at lower layers, protocol header protection for upper layer protocols is provided. This may be sensitive information, in some environments, since it can be used for traffic analysis. Traffic analysis may be countered by a number of means, including message padding (so no information based on message length is exposed), and transmission of dummy messages (so the transmission of real messages is not exposed). Both of these mechanisms assume that traffic is encrypted at some level.

6.6.1.5 When using proprietary network protocols, it is advisable to collapse the model into two layers: the application and the network layer. In this case, it is usually easiest to use application–layer security.

6.6.1.6 Those services which associate data with an originator or recipient (for example, authentication and non repudiation) are best provided at the application layer. This provides the greatest granularity (typically to the individual user). When provided at lower levels, trusted hardware or software is needed to bind the originator to the originating end system. Per-user authentication and non repudiation are recommended for most healthcare applications.

6.6.2 To summarize, placement of security services depends on the proportion (and distribution) of traffic which is considered sensitive according to an organization's security policy. However, some services are only useful at the application layer.

7. Local Security

7.1 When addressing the protection of data in storage, some security services on the end systems are required. Particularly important services include the following: access control, user identification and authentication, and key management.

7.2 *Access Control*:

7.2.1 Having securely transmitted data across a network, protection is necessary from unauthorized disclosure or modification on end systems. Many existing operating systems already provide such access control in conjunction with the logon process. For other platforms, a variety of add–on products are available. Stronger protection from determined adversaries can be provided by encrypting data stored on the local system, particularly when file servers are being used. This topic is addressed in Guides E 1985 and E 1986.

7.2.2 Access control services are, in almost all current systems, implemented and enforced on end systems via the operating system or via application code. Many healthcare applications require more granularity of control (for example, to the field level) than can be provided via the operating system. While some database management systems support this level of granularity, it may be necessary to implement this within the application itself.

7.3 *User Authentication*:

7.3.1 Access control is predicated on proper authentication of the user. A variety of token based authentication products are available to improve on local operating system authentication mechanisms. In some environments, it is necessary to forward authentication information (or evidence of local authentication) to other systems. A number of protocols have been designed to do this, including Kerberos (RFC 1510 and RFC 1964) and SESAME. This topic is discussed (for the centralized case) in Guide E 1985.

7.4 *Protection of Cryptographic Keys*:

7.4.1 It is important to provide secure generation, storage and deletion of keys.

7.4.2 Generation will require a cryptographic quality random number source. This might be hardware (noise diode) or software (cryptographic PRNG).

7.4.3 Keys shall be protected from unauthorized disclosure during their lifetime. Ideally, they would be stored in a separate hardware token (for example, smart card or PCMCIA card) which would also perform cryptographic transformations using the keys. Alternatively, the keys could be stored encrypted under a symmetric or asymmetric key, and decrypted only when needed. For local file encryption, the single key needed could be derived from a password entered by the user, and never stored on the system. See FIPS PUB 140–1 for more details on cryptographic module security.

8. Cryptographic Algorithms and Mechanisms

8.1 This section recommends specific cryptographic algorithms and mechanisms. Recommendations are based on current usage and known security of the algorithms.

8.2 *Algorithms*:

8.2.1 As mentioned previously, the security of a cryptographic algorithm is largely dependent on the size of the keys used. For bulk (symmetric) encryption, keys of 75 bits or longer are appropriate where information must remain secret for extended periods of time. For comparison, see 8.2.1.1-8.2.3.

8.2.1.1 Currently, encryption with key lengths greater than 40 bits cannot be exported from the US (with some exceptions). In early 1997, a brute force search for a 40–bit RC5 key was completed in 3.5 h on a network of 250 workstations.

8.2.1.2 Brute–force search on a 128–bit key using existing technology could not be accomplished in the remaining lifetime of the universe.

8.2.2 The most popular existing standardized algorithm, DES, uses a 56–bit key. In mid–1997, a single DES key was obtained using exhaustive search. This exercise took 4.5 months, and thousands of workstations. While it is premature to say that DES is "broken" (since this type of attack takes a great deal of time and computing resources to obtain a single key), organizations implementing DES should plan to migrate to an alternative algorithm in approximately 5 years. The only standardized replacement is triple DES (DES applied 3 times). This suffers from performance problems when implemented in software (3 times as slow as DES), and suffers from some of the same problems as DES (although not from the "short key" problem). NIST has started the process of selecting a replacement for DES, and it is likely that there will be some idea of what the replacement algorithm will be in 2 to 3 years. Since this is a public process, it is likely that it will be one of the other popular algorithms proposed in recent years (such as IDEA and SAFER–128). Organizations should be wary of selecting an alternative algorithm in the meantime, since there will be interworking problems if another algorithm is standardized, and there will be an enormous amount of analysis of all proposed algorithms, which may expose currently unknown weaknesses.

8.2.3 For asymmetric algorithms, the situation is somewhat easier. All currently popular algorithms are based on computations with very large numbers, and these numbers can simply be made even larger as computational power and cryptanalytic techniques improve. As can be seen in 8.2.4.1-8.2.4.5, most of the common algorithms (such as RSA, Diffie-Hellman, and DSA) require quite long numbers. This is because there are attacks that are "subexponential time" (much faster than a brute–force search, but still dependent on key size). The elliptic curve algorithms discussed below are based on an algebraic system where there are no (known) subexponential attacks, so they can function with much shorter keys for the same strength. However, these algorithms have not been studied for as long as RSA, Diffie–Hellman, etc.

8.2.4 Recommended cryptographic algorithms include the following:

8.2.4.1 *Bulk~~e~~ Encryption*— 2–key or 3–key triple DES in outer–CBC mode (FIPS PUB 46-3, 74, 81, X3.92, and ANSI X9.52).

8.2.4.2 *MAC*—HMAC using SHA1, and HMAC using MD5, RFC 2403, and RFC 2404.

8.2.4.3 *Digital Signature*—RSA as defined in ANSI X9.31 or PKCS #1 (1024–bit key), DSA as defined in ANSI X9.30 or FIPS PUB 186 (1024–bit key), ECDSA as defined in ANSI X9.62 (163–bit key).

8.2.4.4 *Key Management*— Kerberos as defined in RFC 1510 (for small–scale applications), Diffie–Hellman as defined in ANSI X9.42 (1024–bit keys), RSA as defined in ANSI X9.44 (1024–bit keys), elliptic curve versions of Diffie—Hellman as defined in ANSI X9.62 (163–bit keys).

8.2.4.5 *One–way Hash Functions*~~:~~ —SHA-1 (FIPS PUB 180–1).

8.3 *Key Management Mechanisms*:

8.3.1 As noted in 8.2.4.4, Kerberos, which is based on symmetric cryptography, provides encryption and authentication for small environments (up to approximately a few thousand users). There is ongoing work to provide interdomain Kerberos services

TABLE 2 Placement of Security Services

	Confidentiality	Integrity	Entity Authentication	Data Origin Authentication	Non–Repudiation	Access Control and Authorization
Link	SILS	SILS	No standards	SILS	N/A	N/A
Subnetwork	IPSEC	IPSEC	IPSEC	IPSEC	N/A	IPSEC
End–to–end	IPSEC	IPSEC	IPSEC	IPSEC	N/A	IPSEC
Application (session–oriented)	SSL, SPKM	SSL, SPKM	FIPS 196, SPKM	SSL, SPKM	E 1762, PS 100, S-HTTP	PS 103
Application (store–and–forward)	CMS, S/MIME, PGP/MIME, X12.58	CMS S/MIME, PGP/MIME, X12.58	CMS, S/MIME, PGP/MIME, X12.58	CMS, S/MIME, PGP/MIME, X12.58	E 1762, PS 100	PS 103

(for example, between organizations), using public key mechanisms between domains. However, within a single domain, Kerberos requires the use of an online authentication and key distribution server.

8.3.2 Larger scale applications, or those which span organizational boundaries, would do well to use a public–key based protocol. (These protocols do not require an online server, but they do require a CA.) Given the current move toward community health networks and integrated delivery systems, assuming that all traffic will stay within a single organization seems unrealistic. In addition, public key approaches are particularly appropriate for an environment where multiple organizations, which fundamentally do not trust one another, must interact. This is, in large part, due to the fact that the CA can act as a trusted third party (TTP), that is, if all organizations trust the CA, they can trust anyone certified by the CA. There has been much work recently standardizing certificate fields to represent policies, usage constraints, and other mechanisms which can be used to build "trusted certificate paths" between entities. These fields, in effect, allow multiple domains of trust and policy to be overlaid onto a global PKI.

8.3.3 Although this document recommends a variety of public–key based protocols, key management can be simplified by using a single standard certificate format for all protocols. Appropriate standards for certificate management include X.509, X9.57, and X9.55. Most store–and–forward security protocols include the relevant certificates with the protected data. If they are not included in the protocol, certificate retrieval can be done using standardized directory protocols like LDAP (X.500, RFC 1777, RFC 2251, and RFC 2259). Other relevant certificate management documents include the NIST Minimum Interoperability Specification and the ETF Certificate Policy and Certification Practice Statement Framework (RFC 2527). The OCSP protocol defined in RFC 2560 may be used in lieu of CRLs to obtain certificated status. If necessary, ASTM will develop an appropriate certificate profile for healthcare applications.

8.3.4 *Key Recovery*:

8.3.4.1 In some environments which use encryption, there will be a requirement for key recovery so that data may be decrypted if a key is lost or destroyed. This is a requirement for data that is stored in encrypted form. It is not a requirement for data being transmitted over a TCP or similar communications session (using, for example, IPSEC or SPKM), since both parties have access to the unencrypted data on the end systems. It is very likely not a requirement for store–and–forward applications either, although this is dependent on system design.

8.3.4.2 There are a number of proposed mechanisms for key recovery, based on archiving of device–specific keys, archive of the private keys used for key management, use of a trusted escrow agent, etc. Relevant work is going on in NIST and elsewhere.

9. Security Management

9.1 Security management requirements include the following:

9.1.1 Management of security information, such as access control information. This is not an issue for centralized systems, since it is done by the system administrator. For distributed systems, this can be integrated into network management protocols such as OSI CMIP (ISO/IEC 9595 and 9596) and Internet SNMP (RFC 1901–1910).

9.1.2 Audit and archive of security–related information. There are some existing standards for audit and archive (notably ISO/IEC 10164–7 and 10164–8). ASTM Subcommittee E31.17 is working on detailed healthcare–specific requirements for audit and archive.

9.1.3 Ability to activate and deactivate security services. Existing network management protocols such as CMIP and SNMP can be used for this purpose.

9.1.4 Media requirements (for integrity, permanence, and reliability) are being developed by ASTM Committee E-31.

9.1.5 Trusted timestamps are a requirement for many applications. This area is addressed in Guide E 1762. Additional protocol–specific details, if required, will be addressed by other ASTM standards.

10. Existing Standards

10.1 Table 2 illustrates the state of the standards process with respect to existing protocols and applications. Notice the following:

10.1.1 At the link layer, there are few standards (besides algorithm and key management standards). This is tolerable since these are point–to–point connections, so an endpoint only interoperates with one other endpoint.

10.1.2 As discussed in 10.1.1, most current systems implement access control on the end system.

10.1.3 Non repudiation services are generally associated with document or messaging paradigms; CMS, along with S/MIME and other E–mail security protocols provide generic services, while Guide E 1762 and Specification E 2084 accommodate additional requirements at the document level. See Table 2.

11. Keywords

11.1 access control; application security; communications security; cryptography; interoperability; key management; key recovery; local security; security framework; subnetwork security