



Designation: E 2084 – 00

An American National Standard

Standard Specification for Authentication of Healthcare Information Using Digital Signatures¹

This standard is issued under the fixed designation E 2084; 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 specification covers the use of digital signatures to provide authentication of healthcare information, as described in Guide E 1762. It describes how the components of a digital signature system meet the requirements specified in Guide E 1762. This includes specification of allowable signature and hash algorithms, management of public and private keys, and specific formats for keys, certificates, and signed healthcare documents.

1.2 This specification should be read in conjunction with Guide E 1762, which describes the scope of, and requirements for, authentication of healthcare information. This specification describes one implementation (digital signatures) that meets all of the requirements of Guide E 1762. It does not prescribe any particular policy regarding which documents shall be authenticated, and by whom.

2. Referenced Documents

2.1 ASTM Standards:

E 1762 Guide for Electronic Authentication of Healthcare Information²

2.2 ANSI Standards:

X9.30 Part 2: Public Key Cryptography Using Irreversible Algorithms: Secure Hash Algorithm (SHA-1)

X9.31 Reversible Digital Signature Algorithms

X9.55 Extensions to Public Key Certificates and CRLs

X9.57 Certificate Management

X9.62 Elliptic Curve Digital Signature Algorithm

2.3 ISO Standards:

ISO 9594–8 1993: The Directory: Authentication Framework (also available as ITU-S X.509)

ISO 8824–1 1993: Specification of Abstract Syntax Notation One (ASN.1)

ISO 8825–1 1993: Specification of Basic Encoding Rules for ASN.1

ISO 9796 1991: Digital Signature Scheme Giving Message Recovery

ISO 10166 1991: Document Filing and Retrieval (DFR)

2.4 Internet Standards:

RFC 2630 Cryptographic Message Syntax

2.5 Other Documents:

RSA Laboratories, “PKCS #1: RSA Encryption Standard (version 1.5),” November 1993

RSA Laboratories, “PKCS #5: Password Based Encryption (version 1.5),” November 1993

RSA Laboratories, PKCS #6: Extended Certificate Syntax Notation

RSA Laboratories, “PKCS #7: Cryptographic Message Syntax (version 1.5),” November 1993

RSA Laboratories, PKCS #9: Selected Attribute Types

ITU-T X.501 Information Technology Open Systems Interconnection—The Directory: Models

3. Terminology

3.1 Definitions:

3.1.1 *attribute*—piece of information associated with the use of a document.

3.1.2 *authentication (data origin)*—corroboration that this source of data received is as claimed.

3.1.3 *authentication (user)*—provision of assurance of the claimed identity of an entity.

3.1.4 *certificate (public key)*—digitally signed data structure that binds a user’s identity to a public key.

3.1.5 *data integrity*—property that data has not been altered or destroyed in an unauthorized manner.

3.1.6 *digest*—result of applying a one-way hash function to a message.

3.1.7 *digital signature*—data associated with, or a cryptographic transformation of, a data unit that allows a recipient of the data unit to prove the source and integrity of the data unit and protect against forgery, for example, by the recipient.

¹ This specification is under the jurisdiction of ASTM Committee E31 on Healthcare Informatics and is the direct responsibility of Subcommittee E31.25 on Healthcare Management, Security, Confidentiality, and Privacy.

Current edition approved April 10, 2000. Published June 2000.

² Annual Book of ASTM Standards, Vol 14.01.

³ Available from American National Standards Institute, 11 W. 42nd St., 13th Floor, New York, NY 10036.

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

⁵ Available at <http://www.ietf.org>.

⁶ Available from RSA Data Security, 100 Marine Parkway, Redwood City, CA 94605.

3.1.8 *document access time*—time(s) when the subject document was accessed for reading, writing, or editing.

3.1.9 *document attribute*—attribute describing a characteristic of a document. **E 1762**

3.1.10 *document creation time*—time of the creation of the subject document **E 1762**

3.1.11 *document editing time*—time(s) of the editing of the subject document. **E 1762**

3.1.12 *electronic document*—defined set of digital information, the minimal unit of information which may be digitally signed. **E 1762**

3.1.13 *event time*—the time of the documented event.

3.1.14 *(one-way) hash function*—function which maps strings of bits to fixed-length strings of bits, satisfying the following two properties: (1) it is computationally infeasible to find for a given output an input which maps to this output; (2) it is computationally infeasible to find for a given input a second input which maps to the same output.

3.1.15 *private key*—key in an asymmetric algorithm; the possession of this key is restricted, usually to one entity.

3.1.16 *public key*—key in an asymmetric algorithm that is publicly available.

3.1.17 *repudiation*—denial by one of the entities involved in a communication of having participated in all or part of the communication.

3.1.18 *role*—role of a user when performing a signature. Examples include: physician, nurse, allied health professional, transcriptionist/recorder, and others. **E 1762**

3.1.19 *signature attribute*—attribute characterizing a given user's signature on a document **E 1762**

3.1.20 *signature purpose*—indication of the reason an entity signs a document. This is included in the signed information and can be used when determining accountability for various actions concerning the document. Examples include: author, transcriptionist/recorder, and witness. **E 1762**

3.1.21 *signature time*—time a particular signature was generated and affixed to a document. **E 1762**

3.1.22 *signature verification*—process by which the recipient of a document determines that the document has not been altered and that the signature was affixed by the claimed signer. This will in general make use of the document, the signature, and other information such as cryptographic keys or biometric templates.

3.2 *Acronyms: Acronyms:*

3.2.1 *ABA*—American Bar Association

3.2.2 *ASC X9*—Accredited Standards Committee X9

3.2.3 *CA*—Certificate Authority

3.2.4 *CEN*—Comité Européen de Normalisation (European Standards Committee)

3.2.5 *CRL*—Certificate Revocation List

3.2.6 *DSA*—Digital Signature Algorithm

3.2.7 *ISO*—International Standards Organization

3.2.8 *NIST*—National Institute for Standards and Technology

3.2.9 *RSA*—Rivest, Shamir, and Adleman

4. Technology Overview

4.1 Digital signatures are a cryptographic technique in which each user is associated with a pair of keys. The private

key is kept secret, while the public key is distributed to the potential verifiers of the user's digital signature. To sign a document, the document and private key are input to a cryptographic process which outputs a bit string (the signature). To verify a signature, the signature, the document, and the user's public key are input to a cryptographic process, which returns an indication of success or failure. Any modification to the document after it is signed will cause the signature verification to fail (integrity). If the signature was computed using a private key other than the one corresponding to the public key used for verification, the verification will fail (authentication).

4.2 A digital signature is thus a piece of data associated with a document which allows the recipient to prove the origin of the document and to protect against forgery. Digital signatures are formed using asymmetric encryption algorithms as described in 4.1. Digital signatures are associated with a document.

4.3 To sign a message, it is first hashed into a single block using a one-way hash function. A one-way hash function has the property that, given the digest (the output of the hash function), it is computationally infeasible to construct any message that hashes to that value, or to find two messages that hash to the same digest. The digest is processed with the user's private key, and the result may be appended to the message as its signature.

4.4 Separating the signature from the message reduces the amount of data to be encrypted to a single block. This is important since public key algorithms are generally substantially slower than symmetric algorithms. The signature process also introduces redundancy into the message. Redundancy allows the recipient to detect unauthorized changes to the message. Most messages already contain sufficient redundancy to detect such a forgery (such as, English text, time stamps, etc.). The signature process adds additional redundancy, since the message must also hash to the specified digest.

4.5 To verify a signature, the received message is digested, and the digest is processed along with the public key and signature. The result is an indication of success or failure of the verification.

4.6 A digital signature provides the following security services:

4.6.1 Integrity, since any modification of the data being signed will result in a different digest, and thus a different signature.

4.6.2 Origin authentication, since only the holder of the private key corresponding to the public key used for validation could have signed the message.

4.6.3 Support for non-repudiation, that is, irrevocable proof to a third party that only the signer could have created the signature.

4.7 A canonical representation for documents shall be specified for use by cryptographic mechanisms described in this specification. In general, a document may be stored on a system in a different form than the one in which it was generated. For example, if the document contains many numeric values, some systems may store them in a different format.

others as text. Since signatures are computed over representations (encodings), rather than abstract values, there shall be a specific representation the signature is computed over. Such a representation can be defined using Abstract Syntax Notation One (ASN.1) (ISO 8824–1) with an appropriate set of encoding rules, such as the Distinguished Encoding Rules specified in ISO 8825–1. As an added benefit, a number of existing ISO and ANSI standards (notably X.509 and ANSI X9.57) are available that define ASN.1 structures for digital signatures. These are presented in Section 9. The sender and recipient shall agree on an encoding mechanism for the (signed) document. Other mutually agreed or standard or standard encoding mechanisms may be used. If ASN.1 encoding is used, then the specification in Section 9 shall be used. This document representation is taken from ISO 10166.

4.8 A document may be signed by one or more users. Each user's signature and other information is contained in a separate structure. Each signature structure contains an indication of the public key needed to validate the signature and a bit string containing the actual signature. Additionally, other information relevant to the particular signer would be included in an individual signature computation. This per-signer information would be included in the signature computation as *signature attributes*. A signature structure may also include per-signer information which is not signed, but merely appended to the signature structure (*unsigned attributes*). An important unsigned attribute is the *countersignature*. A countersignature is a signature on the signature structure in which it is found, rather than on the document itself. A countersignature thus provides proof of the order in which signatures were applied. Since the countersignature is itself a signature structure, it may itself contain countersignatures; this allows construction of arbitrarily long chains of countersignatures.

5. Mapping to Requirements

5.1 This section maps the algorithms, procedures, and data formats described in Sections 6–9 to the electronic signature requirements presented in Guide E 1762.

5.2 General Requirements:

5.2.1 *Non-repudiation*—This is provided by a digital signature, assuming the private key used for signing is never divulged (see Section 7). Additionally, reliable binding of keys to names shall be provided, using certificates. Appropriate secure archive facilities shall be provided for expired and revoked keys and certificates, as well as CRLs. This might be done by using the new private key to sign the old public key in a certificate.

5.2.2 *Integrity*—This is provided by a digital signature, since any modification of the signed document or signature causes signature verification to fail.

5.3 User Authentication Requirements:

5.3.1 *Secure User Authentication*—This shall be provided by the system implementing digital signature. Secure user authentication is the topic of several standards currently under development by ASTM Committee E31.

5.4 Logical Manifestation Requirements:

5.4.1 *Multiple Signatures*—These are provided, on a document by the **SignerInfo** structures described in Section 9. Additionally, a given signature may have multiple countersignatures.

5.4.2 *Signature Attributes*—These are provided in the **SignerInfo** structure as the **authenticatedAttributes** component.

5.4.3 *Countersignatures*—These are provided using the **countersignature** unsigned attribute.

5.5 Verification Requirements:

5.5.1 *Transportability*—The ASN.1 specification in Section 9 defines a unique encoding for the signed document, which can be transported to another system for verification.

5.5.2 *Interoperability*—Any recipient which implements the ASN.1 specification of Section 9 and the appropriate signature algorithms can verify a signed document.

5.5.3 *Independent Verifiability*—A digital signature is verified using the public key of the signer, which can be certified by a trusted third party (CA) and distributed via a variety of mechanisms.

5.5.4 *Continuity of Signature Capability*—Publication of the public key used for signature verification does not allow an adversary to determine the private key used for signature.

6. Algorithms

6.1 While it is not the intent of this specification to restrict the use of cryptographically sound signature algorithms, it is desirable to limit the number of allowable algorithms, to ensure interoperability. This specification recommends use of the following algorithms:

6.1.1 RSA and variants, as described in ANSI X9.31, PKCS #1, and Appendix X1, or

6.1.2 DSA and variants, as described in ANSI X9.30, ANSI X9.62, and Appendix X2.

6.2 Allowable hash algorithms for use with the signature algorithms include:

6.2.1 For RSA and variants, SHA-1.

6.2.2 For DSA and variants, SHA-1.

7. Public Key Management

7.1 To verify a signature, a user shall obtain the signer's public key from a source that he or she trusts. This source is a public key certificate, which binds a user's name to his public key. Certificates are signed by a trusted issuer, the 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. The format of a certificate is specified in ISO 9594–8 and in ANSI X9.57. Section 9 of this specification describes this format in more detail.

7.2 Although this specification does not impose any particular structure on the CAs, many implementations find it reasonable to impose a hierarchical structure. A hierarchy of CAs can be set up, where the higher level CAs sign the certificates of the CAs beneath them, etc. The lowest level of CAs sign user certificates. At the top of this hierarchy are relatively few CAs (perhaps one per country) who may “cross-certify” each other's public keys.

7.3 Various security architectures define mechanisms to construct a certification path through the hierarchy to obtain a given user's certificate and all CA certificates in the path to

validate it. These architectures share the common characteristics that a user need only trust one other public key in order to obtain and validate any other certificate. The trusted key may be that of the top-level CA (in a centralized trust model), or the local CA that issued the user's certificate (in a decentralized model). Since there will be a variety of entities offering certification services (ranging from commercial entities to the U.S. government), implementations may find it useful to accommodate multiple trusted public keys, in the absence of cross-certification between all of these services.

7.4 Certificates contain an expiration date. If it is necessary to cancel a certificate prior to its expiration date (for example, if the name association becomes invalid or the corresponding private key is lost or compromised), the certificate may be added to the CA's certificate revocation list (CRL). This list is signed by the CA and widely distributed (for example, as part of the CA's directory entry). Each entry contains the revoked certificate's serial number, a revocation time, and optionally a revocation reason and time of suspected compromise. The certificate remains on the CRL until its expiration date. A system will typically archive expired certificates and CRLs in order to be able to verify signatures after the fact. Such archives shall be protected from modification, perhaps by using write-once media such as optical disk, or by digitally signing the archive files. In the latter case, the files may periodically need to be re-signed as the certificates needed to verify the signatures expire.

7.5 Certificates and CRLs for use with this specification are defined in ANSI X9.57 and ANSI X9.55. The relevant ASN.1 specification can be found in Section 9.

8. Private Key Management

8.1 To support a non-repudiation service, the user's private key shall be protected from disclosure to other users. This ensures that only that user could have created a digital signature using the private key; therefore the user cannot repudiate a signature by claiming that someone else applied it.

8.2 The most secure way to protect the private key is to embed it in a tamperproof cryptographic module that will perform the signature computation internally. Such modules might include smart cards, cryptographic diskettes, and PCM-CIA cards. Access to the signature function requires the user to authenticate himself to the module using passwords, PINs, or biometric controls (even including graphic signature verification), or a combination thereof. This approach is recommended by this specification.

8.3 A less secure way to protect the private key is to encrypt it under a secret key computed from a password entered by the user. The minimum allowable key length for such a secret key is 56 bits (the key length for the Data Encryption Standard). Use of the password also authenticates the user to the signing system. (One such mechanism is described in PKCS #5.) The encrypted private key may be stored on removable media like a floppy disk and decrypted when needed to perform a signature. This approach is acceptable, but not recommended by this specification.

9. ASN.1 Specification

9.1 Following is the ASN.1 specification for the data structures defined in this specification. Complete specifications for ASN.1 can be found in ISO 8824-1 and 8825-1.

9.2 Document Structure and Attributes:

9.2.1 A document is represented using the following ASN.1 type. This representation is only required when computing or verifying a signature on the document, and places no constraints on how the document is represented during transmission or storage (although ASN.1 might also be appropriate for these purposes). Attributes may be specified, if desired, using the ATTRIBUTE class that follows, or that in ITU-T X.501.

```
Document ::= SEQUENCE {
    attributes      SET OF Attribute,
    content         CHOICE {
        digest      DigestDocument,
        full        OCTET STRING }
}

Attribute ::= SEQUENCE {
    type  ATTRIBUTE.&id ({SupportedAttributes}),
    values SET OF ATTRIBUTE.&Syntax ({SupportedAttributes} {@type})
}

ATTRIBUTE CLASS ::= {
    &Syntax,
    &singleValued          BOOLEAN DEFAULT FALSE,
    &id                     OBJECT IDENTIFIER UNIQUE
}

WITH SYNTAX {
    WITH ATTRIBUTE SYNTAX&Syntax
    [SINGLE VALUE          &singleValued]
    ID                    &id
}

DigestDocument ::= SEQUENCE {
    algorithm      AlgorithmIdentifier,
}

DigestDocument ::= SEQUENCE {
    algorithm      AlgorithmIdentifier,
    digest          OCTET STRING,
    locator        DocumentLocator OPTIONAL
}

DocumentLocator ::= CHOICE {
    uri            [0]      IA5String,
    dfr            [1]      DFRName, --using "uri" attribute
    oid            [2]      OBJECT IDENTIFIER,
    other          [3]      INSTANCE OF OTHER-LOCATOR
}

DFRName ::= SEQUENCE {
    docStore      AETitle,
    uri            OCTET STRING
}

AETitle ::= CHOICE {
    nameForm      Name, -- last RDN is "qualifier"
    oidForm       SEQUENCE {
        title      OBJECT IDENTIFIER,
        qualifier   INTEGER
    }
}

OTHER-LOCATOR ::= TYPE-IDENTIFIER
```

9.2.2 This structure specifies the physical representation of a document when presented to the authorization mechanism. This does not imply that the document shall be stored, transmitted, or otherwise manipulated using this representation at any time other than authorization processing. However, it may be efficient to store the attributes and digest as an ASN.1 encoded structure.

9.2.3 The **digest** alternative contains the document attributes and the digest of the document, along with an optional document locator. The **full** alternative contains the actual document content, as an OCTET STRING.