DICOM Change Proposal

STATUS	Assigned
Date of Last Update	2025/12/10
Person Assigned	Robert Horn, rjhorniii@mail.com
Submitter Name	Essien Ge, essien.ge@philips.com
Submission Date	2025/11/04

Change Number	CP-2596
Log Summary: Update digital signatures and hash algorithms. Clarify MAC vs digest terminology usage.	
Name of Standard PS3.3, PS3.15	
Rationale for Change: 1. At the time when the current profile was published, the difference between the message authentication code (MAC) and message digest was not clearly defined. The MAC uses an algorithm to generate an authentication code with a symmetric key (See definition in ISO/IEC 9797-1), while message digest is the result of a hash. In most signatures, a message digest is used. Update relevant descriptions to use current terminology from ISO and NIST. This affects 3 Macros in PS3.3, *C.12.1.1.3 Digital Signatures Macro, C.17.2.1 Hierarchical SOP Instance Reference Macro, C.38.2.2 Stored File Access Macro.* ***Issue 1: A MAC requires the independent exchange of a shared secret key. We did not disallow this in the original definition of Referenced SOP Instance MAC Sequence (0400,0403). Can we disallow it in this CP? Alternatively, we can use a note that indicates that this will almost always be a message digest because use of a MAC requires an independent exchange of a secret key? (This will also be a WG-06 issue, but WG-14 should have an opinion to give WG-06.)*** ***Issue 2: Do we add the notes to the definitions? DICOM does not usually discuss this kind of issue. It's usually left to the issuer of the definition to update this kind of information, in this case they are issued by IETF, NIST, and ISO.***	
Change Wording:	

Update PS3.3 section 2 Normative References as indicated below

2 Normative References

The following standards contain provisions that, through reference in this text, constitute provisions of this Standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this Standard are encouraged to investigate the possibilities of applying the most recent editions of the standards indicated below.

…

[NIST 800-107r1] NIST Special Publication 800-107, Revision 1, Recommendation for Applications Using Approved Hash Algorithms, https://doi.org/10.6028/NIST.SP.800-107r1

[NIST 800-131Ar2] NIST SP 800-131A Rev. 2 Transitioning the Use of Cryptographic Algorithms and Key Lengths. https://doi.org/10.6028/NIST.SP.800-131Ar2

[NIST SP 800-133r2] Recommendation for Cryptographic Key Generation, https://doi.org/10.6028/NIST.SP.800-133r2

Update PS3.3 Section 3.15 Security Definitions

3.15 SECURITY DEFINITIONS

This Part of the Standard makes use of the following terms defined in [ECMA 235]:

Security Context The definition is "security information that represents, or will represent a Security Association to an initiator or acceptor that has formed, or is attempting to form such an association."

This Part of the Standard makes use of the following terms defined in [NIST 800-133r2r1]:

Message Authentication Code See [NIST 800-133r2]

Note: The definition is "A cryptographic checksum on data that uses an approved security function and a symmetric key to detect both accidental and intentional modifications of data."

This Part of the Standard makes use of the following terms defined in [NIST 800-107r1]:

Message Digest See [NIST 800-107r1]

Note: The definition is "The result of applying a hash function to a message. Also known as a 'hash value' or 'hash output'."

Update PS3.3 Section 3.16 DICOM Security Profiles

3.16 DICOM SECURITY PROFILES

This Part of the Standard makes use of the following terms defined in PS3.15:

~~**Message Authentication Code (MAC) See Message Authentication Code in PS3.15.**~~

Certificate See Certificate in PS3.15.

Kommentiert [EG1]: Update the terminology in PS3.3 by refering to NIST 800-133 instead of the current definition in PS3.15, which is also replaced.
Add term Message digest (also know as hash value or hash output) as defined in NIST 800-107

Update PS3.3 Annex C.12.1.1.3 Digital Signatures Macro

C.12.1.1.3 Digital Signatures Macro

This Macro allows Digital Signatures to be included in a DICOM Data Set for the purpose of ~~**insuring**~~ **ensuring** the integrity of the Data Set, and to authenticate the sources of the Data Set. Table C.12-6 defines the Attributes needed to embed a Digital Signature in a Data Set. This Macro may appear in individual Sequence Items as well as in the top level Data Set of the SOP Instance.

Note

1. Each Item of a Sequence of Items is a Data Set. Thus, individual Sequence Items may incorporate their own Digital Signatures in addition to any Digital Signatures added to the Data Set in which the Sequence appears.
2. The inclusion of this Macro in Sequence Items, other than as specified in this Part of the Standard, may be specified in an application-defined Standard Extended SOP Class or Private SOP Class (see PS3.2).

The process of generating a DICOM digital signature begins with the calculation of a Message Digest of the specified data elements using a cryptography hash algorithm, which is subsequently signed with a digital signature algorithm.

Table C.12-6. Digital Signatures Macro Attributes

Attribute Name	Tag	Type	Attribute Description
MAC Parameters Sequence	(4FFE,0001)	3	A Sequence of Items that describe the parameters used to calculate a ~~**MAC**~~ **Message Digest** for use in Digital Signatures. One or more Items shall be included in this Sequen **Note** **While the attribute MAC Parameter Sequ (4FFE,0001) can be used to carry a Mess Authentication Code (MAC) or Message Digest, a Message Digest is normally use digital signature purposes. A Message Authentication Code (MAC) requires a separately exchanged secret key.**
>MAC ID Number	(0400,0005)	1	A number, unique within this SOP Instance, used to identify this MAC Parameters Sequence (4FFE,000 Item from an Item of the Digital Signatures Sequenc (FFFA,FFFA).
>MAC Calculation Transfer Syntax UID	(0400,0010)	1	The Transfer Syntax UID used to encode the values the Data Elements included in the ~~**MAC**~~ **Message Digest** calculation. Only Transfer Syntaxes that exp include the VR and use Little Endian encoding shall used. **Note** **Certain Transfer Syntaxes, particularly th that are used with compressed data, allo fragmentation of the pixel data to change. If such fragmentation changes, Digital Signatures generated with such Transfer Syntaxes could become invalid.**
> MAC Algorithm	(0400,0015)	1	The algorithm used in generating the ~~**MAC**~~ **Message Digest** to be ~~**encrypted**~~ **signed** to form the Digital Signature. For Defined Terms, see Table C.12.1.1.3.1.2-1, "De Terms for MAC Algorithm (0400,0015)".
>Data Elements Signed	(0400,0020)	1	A list of Data Element Tags in the order **that** they appear in the Data Set that identify the Data Elements used in creating the ~~**MAC**~~ **Message Digest** for the Digital Signature. See Section C.12.1.1.3.1.1.
Digital Signatures Sequence	(FFFA,FFFA)	3	Sequence holding Digital Signatures. One or more Items are permitted in this Sequence.
> MAC ID Number	(0400,0005)	1	A number used to identify which MAC Parameters Sequence Item was used in the calculation of this Digital Signature.
>Digital Signature UID	(0400,0100)	1	A UID that can be used to uniquely reference this signature.

Kommentiert [EG2]: Updated to "MAC or digest" according to the discussion in WG14 meeting on Oct.9.

However still struggling here as:

- all the algorithm allowed in **Table C.12.1.1.3.1.2-1. Defined Terms for MAC Algorithm (0400,0015)** are hash algorithm not MAC algorithm (e.g. HMAC, CMAC).
- the current DICOM spec cannot address the manage of MAC key for such case, no attribute allow this, which makes it in practice not possible to use MAC instead of a digest.
- last but not least, nowadays all the cryptography libs are doing the sign/verify operation with the hash function integrated, means people are not possible to replace it with MAC unless the want to use a self-written crypto lib.

Kommentiert [EG2R2]: Update description to Message Digest.
Add a note saying that the attribute can carry both MAC and digest, but here digest is used.

Kommentiert [EG3]: Use signed instead of encrypted. It maybe correct to say encrypted for RSA digital signature, but not accurate for dedicate digital signature algorithm such as DSA, ECDSA or ML-DSA.

Attribute Name	Tag	Type	Attribute Description
>Digital Signature DateTime	(0400,0105)	1	The date and time the Digital Signature was created. The time shall include an offset (i.e., time zone indication) from Coordinated Universal Time. Note This is not a certified timestamp, and hence is not completely verifiable. An application can compare this date and time with those of other signatures and the validity date of the certificate to gain confidence in the veracity of this date and time.
>Certificate Type	(0400,0110)	1	The type of certificate used in (0400,0115). Defined Terms: **X509_1993_SIG** Note Digital Signature Security Profiles (see PS3.15) may require the use of a restricted subset of these terms.
>Certificate of Signer	(0400,0115)	1	A certificate that holds the identity of the entity producing this Digital Signature, that entity's public key or key identifier, and the algorithm and associated parameters with which that public key is to be used. Algorithms allowed are specified in Digital Signature Security Profiles (see PS3.15). Note 1. As technology advances, additional encryption algorithms may be allowed in future releases. Implementations should take this possibility into account. 2. When symmetric encryption is used, the certificate merely identifies which key was used by which entity, but not the actual key itself. Some other means (e.g., a trusted third party) must be used to obtain the key.
>Signature	(0400,0120)	1	The ~~**MAC**~~ **Message digest** generated as described in Section C.12.1.1.3.1.1 and ~~**encrypted**~~**signed** using the algorithm, parameters, and private key associated with the Certificate of the Signer (0400,0115). See Section C.12.1.1.3.1.2.
>Certified Timestamp Type	(0400,0305)	1C	The type of certified timestamp used in Certified Timestamp (0400,0310). Required if Certified Timestamp (0400,0310) is present. Defined Terms: **CMS_TSP** Internet X.509 Public Key Infrastructure Time Stamp Protocol Note Digital Signature Security Profiles (see PS3.15) may require the use of a restricted subset of these terms.
>Certified Timestamp	(0400,0310)	3	A certified timestamp of the Digital Signature (0400,0120) Value, which shall be obtained when the Digital Signature is created. See Section C.12.1.1.3.1.3.

Attribute Name	Tag	Type	Attribute Description
>Digital Signature Purpose Code Sequence	(0400,0401)	3	The purpose of this Digital Signature. Only a single Item is permitted in this Sequence.
>>Include Table 8.8-1 "Code Sequence Macro Attributes"			*BCID 7007 "Signature Purpose".*

C.12.1.1.3.1 Digital Signatures Macro Attribute Descriptions

C.12.1.1.3.1.1 Data Elements Signed

The Data Elements Signed Attribute shall list the Tags of the Data Elements that are included in the ~~**MAC**~~ **Message Digest** calculation. The Tags listed shall reference Data Elements at the same level as the ~~**Mac**~~**MAC** Parameters Sequence (4FFE,0001) Data Element in which the Data Elements Signed Attribute appears. Tags included in Data Elements Signed shall be listed in the order in which they appear within the Data Set.

The following Data Elements shall not be included either implicitly or explicitly in the list of Tags in Data Elements Signed, nor included as part of the ~~**MAC**~~ **Message Digest** calculation:

- The Length to End (0008,0001) or any Tag with an element number of 0000 (i.e., no Data Set or group lengths may be included in ~~**MAC**~~ **Message Digest** calculations)
- Tags with a group number less than 0008
- Tags associated with Data Elements whose VR is UN
- Tags of Data Elements whose VR is SQ, where any Data Element within that Sequence of Items has a VR of UN recursively
- Tags with a group number of FFFA (e.g., the Digital Signatures Sequence)
- MAC Parameters Sequence (4FFE,0001)
- Data Set Trailing Padding (FFFC,FFFC)
- Item Delimitation Item (FFFE,E00D)

Note

1. The Length to End and group lengths can change if non-signed Data Elements change, so it is not appropriate to include them in the ~~**MAC**~~ **Message Digest** calculation.
2. Since the Data Element Tags that identify a Sequence and the start of each Item are included in the ~~**MAC**~~ **Message Digest** calculation, there is no need to include the Item Delimitation Item Tags.

If any of the Data Element Tags in the list refer to a Sequence of Items, then the Tags of all Data Elements within all Items of that Sequence shall be implicitly included in the list of Data Elements Signed, except those disallowed above. This implicit list shall also include the Item Tag (FFFE,E000) Data Elements that separate the Sequence Items and the Sequence Delimitation Item (FFFE,E0DD).

Note

It is possible to sign individual Items within a Sequence by including the Digital Signatures Macro in that Sequence Item. In fact, this is a highly desirable feature, particular when used in the context of reports. The Digital Signatures Macro is applied at the Data Set level, and Sequences of Items are merely Data Sets embedded within a larger Data Set. Essentially, the Digital Signatures Macro may be applied recursively.

An example of nesting Digital Signatures within Data Elements is illustrated in Figure C.12-1.

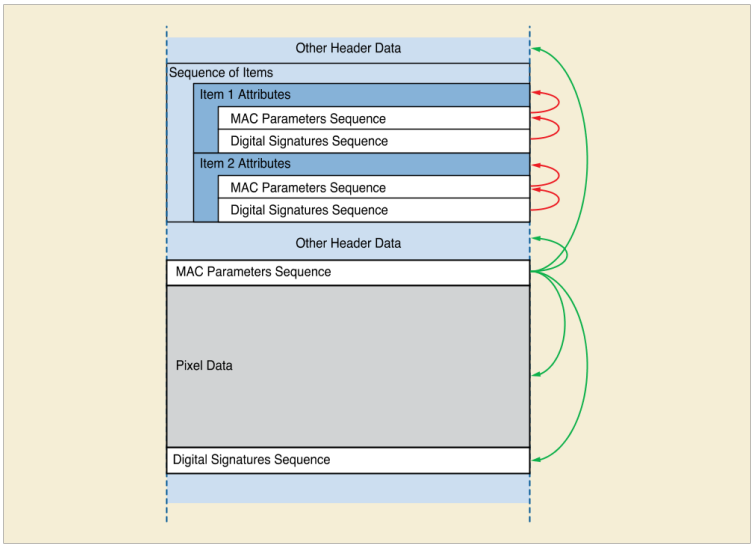

Figure C.12-1. Example of Nesting Digital Signatures (Informative)

In this example, there is main signature covering the pixel data and a few other Data Elements, plus two individually signed Items within a Sequence.

For Data Elements with a VR OB (e. g. pixel data) that have an undefined length (i.e., the data is encapsulated as described in PS3.5), the Item Data Element Tags that separate the fragments shall implicitly be included in the list of Data Elements Signed (i.e., a Data Element with a VR of OB is encoded in the same fashion as a Sequence of Items).

C.12.1.1.3.1.2 Signature

To generate the ~~MAC~~ **Message Digest** , Data Elements referenced either explicitly or implicitly by the Tags in the Data Elements Signed (0400,0020) list shall be encoded using the Transfer Syntax identified by the MAC Calculation Transfer Syntax UID (0400,0010) of the MAC Parameters Sequence (0400,0010) Item where the Data Elements Signed (0400,0020) appears. Data shall be formed into a byte stream and presented to the algorithm specified by MAC Algorithm (0400,0015) for computation of the ~~MAC~~ **Message Digest** according to the following rules:

For all Data Elements except those with a VR of SQ or with a VR of OB with an undefined length, all Data Element fields, including the Tag, the VR, the reserved field (if any), the Value Length, and the Value, shall be placed into the byte stream in the order encountered.

For Data Elements with a VR of SQ or with a VR of OB with an undefined length, the Tag, the VR, and the reserved field are placed into the byte stream. The Value Length shall not be included. This is followed by each Item Tag in the order encountered, without including the Value Length, followed by the contents of the Value for that Item. In the case of an Item within a Data Element whose VR is SQ, these rules are applied recursively to all of the Data Elements within the Value of that Item. After all the Items have been incorporate into the byte stream, a Sequence Delimitation Item Tag (FFFE,E0DD) shall be added to the byte stream presented to the MAC Algorithm **(0400, 0015)** , regardless of whether or not it was originally present.

Note

Since the Value Length of Data Elements with a VR of SQ can be either explicit or undefined, the Value Lengths of such Data Elements are left out of the MAC **or Digest** calculation. Similarly, the Value Length of Data Elements with a VR of OB with an undefined length are also left out so that they are handled consistently. If such Data Elements do come with undefined lengths, including the Item Tags that separate the Items or fragments insures that Data Elements cannot be moved between Items or Fragments without compromising the Digital Signature. For those Data Elements with explicit lengths, if the length of an Item changes, the added or removed portions would also impact the MAC **or Digest** calculation, so it is not necessary to include explicit lengths in the MAC calculation. It is possible that including the Value Lengths could make cryptanalysis easier.

After the fields of all the Data Elements in the Data Elements Signed list have been placed into the byte stream presented to the MAC Algorithm **(0400, 0015)** according to the above rules, all of the Data Elements within the Digital Signatures Sequence Item except the Certificate of Signer (0400,0115), Signature (0400,0120), Certified Timestamp Type (0400,0305), and Certified Timestamp (0400,0310) shall also be encoded according to the above rules, and presented to the MAC algorithm **(0400, 0015)** (i.e., the Attributes of the Digital Signature Sequence Item for this particular Digital Signature are also implicitly included in the list of Data Elements Signed, except as noted above).

The resulting ~~**MAC**~~ **Message Digest** code after processing this byte stream by the MAC Algorithm **(0400, 0015)** is then encrypted as specified in the Certificate of Signer and placed in the Value of the Signature Data Element.

Note

1. The Transfer Syntax used in the ~~**MAC**~~ **Message Digest** calculation may differ from the Transfer Syntax used to exchange the Data Set.
2. Digital Signatures require explicit VR in order to calculate the ~~**MAC**~~ **Message Digest** . An Application Entity that receives a Data Set with an implicit VR Transfer Syntax may not be able to verify Digital Signatures that include Private Data Elements or Data Elements unknown to that Application Entity. This also true of any Data Elements whose VR is UN. Without knowledge of the Value Representation, the receiving Application Entity would be unable to perform proper byte swapping or be able to properly parse Sequences in order to generate a MAC.
3. If more than one entity signs, each Digital Signature would appear in its own Digital Signatures Sequence Item. The Digital Signatures may or may not share the same MAC Parameters Sequence **(4FFE,0001)** Item.
4. The notion of a notary public (i.e., someone who verifies the identity of the signer) for Digital Signatures is partially filled by the authority that issued the Certificate of Signer.

Table C.12.1.1.3.1.2-1 lists the Defined Terms for MAC Algorithm (0400,0015).

Table C.12.1.1.3.1.2-1. Defined Terms for MAC Algorithm (0400,0015)

Defined Term	Reference
RIPEMD160	[ISO/IEC 10118-3]
MD5	[RFC1321]
SHA1	[FIPS PUB 180-4]
SHA224	[FIPS PUB 180-4]
SHA256	[FIPS PUB 180-4]
SHA384	[FIPS PUB 180-4]
SHA512	[FIPS PUB 180-4]
SHA512_224	[FIPS PUB 180-4]
SHA512_256	[FIPS PUB 180-4]
SHA3_224	[FIPS PUB 202]
SHA3_256	[FIPS PUB 202]
SHA3_384	[FIPS PUB 202]
SHA3_512	[FIPS PUB 202]

Note

Security Profiles (see PS3.15) may restrict or extend the list of MAC algorithms that are permitted or required by a specific profile.

C.12.1.1.3.1.3 Certified Timestamp

To generate a certified timestamp, the Value of the Signature (0400,0120) is transmitted to a third party, as specified by the protocol referred to by the Certified Timestamp Type (0400,0305). The third party then generates and returns a certified timestamp in the form specified by that protocol. The certified timestamp returned by the third party is encoded as a stream of bytes in the Certified Timestamp Attribute.

Note

The timestamp protocol may be specified by a Profile in PS3.15.

Update PS3.3 C.17.2.1 Hierarchical SOP Instance Reference Macro

C.17.2.1 Hierarchical SOP Instance Reference Macro

Table C.17-3 specifies the Attributes of the Hierarchical SOP Instance Reference Macro, which reference**s** a list of SOP Instances.

Table C.17-3. Hierarchical SOP Instance Reference Macro Attributes

Attribute Name	Tag	Type	Attribute Description
Study Instance UID	(0020,000D)	1	Unique identifier for the Study.
Referenced Series Sequence	(0008,1115)	1	Sequence of Items where each Item includes the Attributes of a Series containing referenced Composite Object(s). One or more Items shall be included in this Sequence.
>Include Table C.17-3a "Hierarchical Series Reference Macro Attributes"			

Table C.17-3a specifies the Attributes that reference a Series of SOP Instances.

To protect integrity of the referenced SOP instances, DICOM uses either a Digital Signature carried in Referenced Digital Signature Sequence (0400,0402) or a Message Digest carried in Referenced SOP Instance MAC Sequence (0400,0403).

Table C.17-3a. Hierarchical Series Reference Macro Attributes

Attribute Name	Tag	Type	Attribute Description
Series Instance UID	(0020,000E)	1	Unique identifier of a Series that is part of this Study and contains the referenced Composite Object(s).
Retrieve AE Title	(0008,0054)	3	Title of the DICOM Application Entity where the Composite Object(s) may be retrieved on the network.
Retrieve Location UID	(0040,E011)	3	Unique identifier of the system where the Composite Object(s) may be retrieved on the network.
Retrieve URL	(0008,1190)	3	URL specifying the location of the referenced Instance(s).
Storage Media File-Set ID	(0088,0130)	3	The user or implementation specific human readable identifier that identifies the Storage Media on which the Composite Object (s) reside.

Attribute Name	Tag	Type	Attribute Description
Storage Media File-Set UID	(0088,0140)	3	Uniquely identifies the Storage Media on which the Composite Object(s) reside.
Referenced SOP Sequence	(0008,1199)	1	References to Composite Object SOP Class/SOP Instance pairs that are part of the Study defined by Study Instance UID and the Series defined by Series Instance UID (0020,000E). One or more Items shall be included in this Sequence.
>Include Table 10-11 "SOP Instance Reference Macro Attributes"			
>Purpose of Reference Code Sequence	(0040,A170)	3	Describes the purpose for which the reference is made. One or more Items are permitted in this Sequence.
>>Include Table 8.8-1 "Code Sequence Macro Attributes"			*Baseline CID may be specified in Macro invocation.*
>Referenced Digital Signature Sequence	(0400,0402)	3	Sequence of references to Digital Signatures in the referenced SOP Instance. One or more Items are permitted in this Sequence. Note The Attributes in this Sequence can be used to detect if the referenced SOP Instance has been altered.
>>Digital Signature UID	(0400,0100)	1	The Unique Identifier of a Digital Signature held in the referenced SOP Instance.
>>Signature	(0400,0120)	1	The Signature Value identified by the Digital Signature UID within the Referenced SOP Instance UID.
>Referenced SOP Instance MAC Sequence	(0400,0403)	3	A ~~**MAC**~~**Message Digest** Calculation from data in the referenced SOP Instance that can be used as a data integrity check. Only a single Item is permitted in this Sequence. Note This Attribute may be used in place of Referenced Digital Signature Sequence (0400,0402), particularly if the SOP Instance does not have appropriate Digital Signatures that can be referenced.
>>MAC Calculation Transfer Syntax UID	(0400,0010)	1	The Transfer Syntax UID used to encode the values of the Data Elements included in the ~~**MAC**~~**Message Digest** calculation. When computing the ~~**MAC**~~**Message Digest**, only Transfer Syntaxes that explicitly include the VR and use Little Endian encoding shall be used. Note 1. Certain Transfer Syntaxes, particularly those that are used with compressed data, allow the fragmentation of the pixel data to change. If such fragmentation changes, Digital Signatures generated with such Transfer Syntaxes could become invalid. 2. This does not constrain the Transfer Syntax used to transmit the object.
>>MAC Algorithm	(0400,0015)	1	The algorithm used in generating the ~~**MAC**~~**Message Digest**.

Attribute Name	Tag	Type	Attribute Description
			For Defined Terms, see Table C.12.1.1.3.1.2-1, “Defined Terms for MAC Algorithm (0400,0015)”.
>>Data Elements Signed	(0400,0020)	1	A list of Data Element Tags in the order they appear at the top level of the referenced SOP Instance that identify the Data Elements used in creating the ~~**MAC**~~**Message Digest**. See Section C.12.1.1.3.1.1.
>>MAC	(0400,0404)	1	The ~~**MAC**~~**Message Digest** generated as described in Section C.12.1.1.3, but unencrypted and without inclusion of fields from the Digital Signatures Sequence. See Section C.12.1.1.3.1.2.

Update PS3.3 C.38.2.2 Stored File Access Macro

C.38.2.2 Stored File Access Macro

175 Table C.38.2-2 specifies the Attributes of the Stored File Access Macro, which describe non-DICOM protocol access to a stored SOP Instance in the DICOM File Format, possibly contained in a container file.

Table C.38.2-2. Stored File Access Macro Attributes

Attribute Name	Tag	Type	Attribute Description
File Access URI	(0008,0409)	1C	Access URI for a file containing the SOP Instance. See Section C.38.2.2.1.1. Required if the referenced SOP Instance is in the DICOM File Format, and is accessible through a non-DICOM protocol (see Annex P).
Container File Type	(0008,040A)	1C	Type of container file. See Section C.38.1.2.8 for Defined Terms. Required if File Access URI (0008,0409) is present.
Filename in Container	(0008,040B)	1C	Filename within a container file of the file containing the SOP Instance. See Section C.38.2.2.1.2. Required if Container File Type (0008,040A) is ZIP, TAR, or TARGZIP.
File Offset in Container	(0008,040C)	1C	Byte offset (zero-based) within a container file for the start of the SOP Instance file. See Section C.38.2.2.1.2. Required if Container File Type (0008,040A) is BLOB. May be present otherwise.
File Length in Container	(0008,040D)	1C	Length in bytes of the SOP Instance file within a container file. See Section C.38.2.2.1.2. Required if Container File Type (0008,040A) is BLOB. May be present otherwise.
Stored Instance Transfer Syntax UID	(0008,040E)	1C	Transfer Syntax of the SOP Instance encoded in DICOM File Format. Equal to Transfer Syntax UID (0002,0010) in File Meta Information header of the stored Instance. Required if File Access URI (0008,0409) is present.
Lossy Image Compression Ratio	(0028,2112)	3	Describes the approximate lossy compression ratio(s) that have been applied to this image.

Attribute Name	Tag	Type	Attribute Description
			See Section C.7.6.1.1.5.2.
MAC Algorithm	(0400,0015)	1C	The algorithm used for generating a ~~**Message Authentication Code**~~**Message Digest**. See Table C.12.1.1.3.1.2-1 for Defined Terms. Required if MAC (0400,0404) is present.
MAC	(0400,0404)	3	~~**Message Authentication Code**~~**Message Digest** computed across the stored instance file for verification of file integrity. See Section C.38.2.2.1.3.

C.38.2.2.1 Stored File Access Macro Attribute Descriptions

…

C.38.2.2.1.3 MAC (0400,0404)

The integrity of a stored SOP Instance file may be verified by a ~~**Message Authentication Code (also known as a message digest, hash, or cryptographic checksum)**~~ **Message Digest** computed across the SOP Instance file. **The Message Digest is carried using MAC (0400, 0404).** The MAC (0400,0404) Value is computed across the entire file as a byte stream, including the Preamble, Prefix, Meta-Information Header, and Data Set Trailing Padding (see Section 7 "DICOM File Format" in PS3.10). For files stored in container files, the MAC **(0400,0404)** is computed on the file extracted from the container.

Note

1. This differs from the **Message Digest** for Digital Signatures (see Section C.12.1.1.3), which is computed across an enumerated list of Attributes within the SOP Instance, not across the entire file.

Update PS3.15 3.4 Security Definitions

3.4 Security Definitions

This Part of the Standard makes use of the following terms defined in [ECMA 235] :

Security Context See [ECMA 235].

Note The definition is "security information that represents, or will represent a Security Association to an initiator or acceptor that has formed, or is attempting to form such an association."

This Part of the Standard makes use of the following terms defined in [NIST 800-133r2]:

Message Authentication Code See [NIST 800-133r2]

Note The definition is "A cryptographic checksum on data that uses an approved security function and a symmetric key to detect both accidental and intentional modifications of data."

This Part of the Standard makes use of the following terms defined in [NIST 800-107r1]:

Message digest See [NIST 800-107r1]

Note The definition is "The result of applying a hash function to a message. Also known as a 'hash value' or 'hash output'."

Update PS3.15 3.10 DICOM Security Profile Definitions

3.10 DICOM Security Profile Definitions

The following definitions are commonly used in this Part of the DICOM Standard:

Secure Transport Connection A Transport Connection that provides some level of protection against tampering, eavesdropping, **and** masquerading.

~~**Message Authentication Code A digest or hash code derived from a subset of Data Elements.**~~

Certificate An electronic document that identifies a party and that party's public encryption algorithm, parameters, and key. The Certificate also includes, among other things, the identity and a digital signature from the entity that created the certificate. The content and format of a Certificate are defined by ITU-T Recommendation X.509.

Kommentiert [EG4]: Update definition in DICOM PS3.15 to refer to NIST 800-133 definition, in replacement of the original self-defined term.
Add term Message digest (also know as hash value or hash output) as defined in NIST 800-107

Update PS3.15 Section 6.3 Digital Signature Profile

6.3 Digital Signature Profile

An implementation may claim conformance to one or more Digital Signature Profiles.

A Digital Signature profile consists of the following information:

a. The role that the Digital Signature plays, including:

   1. Who or what entity the Digital Signature represents.
   2. A description of the purpose of the Digital Signature.
   3. The conditions under which the Digital Signature is included in the Data Set.

b. A list of Attributes that shall be included in the Digital Signature.

c. The mechanisms that shall be used to generate or verify the Digital Signature, including:

   1. The algorithm and relevant parameters that shall be used to create the MAC or hash code, including the Value to be used for the MAC Algorithm (0400,0015) Attribute.
   2. The ~~**encryption**~~**signature** algorithm and relevant parameters that shall be used to ~~**encrypt**~~**sign** ~~**MAC or hash code**~~**Message Digest** in forming the Digital Signature.
   3. The certificate type or key distribution mechanism that shall be used, including the Value to be used for the Certificate Type (0400,0110) Attribute.
   4. Any requirements for the Certified Timestamp Type (0400,0305) and Certified Timestamp (0400,0310) Attributes.

d. Any special requirements for identifying the signatory.

e. The relationship with other Digital Signatures, if any.

f. Any other factors needed to create, verify, or interpret the Digital Signature

Digital Signature Profiles are specified in Annex C.

Update PS3.15 Annex C Digital Signature Profiles(Normative)

C Digital Signature Profiles (Normative)

C.1 Base RSA Digital Signature Profile

The Base RSA Digital Signature Profile outlines the use of RSA **to sign** ~~**encryption of**~~ a ~~**MAC**~~**Message Digest** to generate a Digital Signature. This Profile does not specify any particular set of Data Elements to sign. Other Digital Signature profiles may refer to this profile, adding specifications of which Data Elements to sign or other customizations.

The creator of a digital signature shall use one of the RIPEMD-160, MD5, SHA-1 ~~**or,**~~ SHA-2 family (SHA256, SHA384, SHA512, of hashing functions to generate a ~~**MAC**~~ **Message Digest** , which is then ~~**encrypted**~~**signed** using a private RSA key. All validators of digital signatures shall be capable of using a ~~**MAC**~~ **Message Digest** generated by any of the hashing functions specified (RIPEMD-160, MD5, SHA-1 or SHA256, SHA384, SHA512.

Note ~~**The use of MD5 is not recommended by its inventors, RSA. See:ftp://ftp.rsasecurity.com/pub/pdfs/bulletn4.pdf**~~ **The use of RIPEMD160, MD5, and SHA1 are subject to use recommendations by NIST and others that limit use for creating new signatures. [NIST SP 800-131A Revision 2]**

The data to be signed shall be padded to a block size matching the RSA key size, as directed in [RFC 2437] (PKCS #1). The Value of MAC Algorithm (0400,0015) shall be set to either "RIPEMD160", "MD5", "SHA1", "SHA256", "SHA384" ~~**or,**~~ "SHA512"). The public key associated with the private key as well as the identity of the Application Entity or equipment manufacturer that owns the RSA key pair shall be transmitted in an [ITU-T X.509] (1993) signature certificate. The Value of the Certificate Type (0400,0110) Attribute shall be set to "X509_1993_SIG". A site-specific policy determines how the [ITU-T X.509] certificates are generated, authenticated, and distributed. A site may issue and distribute [ITU-T X.509] certificates directly, may utilize the services of a Certificate Authority, or use any reasonable method for certificate generation and verification.

If an implementation utilizes timestamps, it shall use a Certified Timestamp Type (0400,0305) of "CMS_TSP". The Certified Timestamp (0400,0310) shall be generated as described in [RFC 3161].

C.2 Creator RSA Digital Signature Profile

The creator of a DICOM SOP Instance may generate signatures using the Creator RSA Digital Signature Profile. The Digital Signature produced by this Profile serves as a lifetime data integrity check that can be used to verify that the pixel data in the SOP instance has not been altered since its initial creation. An implementation that supports the Creator RSA Digital Signature Profile may include a Creator RSA Digital Signature with every SOP Instance that it creates; however, the implementation is not required to do so.

The signature shall use one of the RIPEMD-160, MD5, SHA-1 or SHA-2 family (SHA256, SHA384, SHA512) of hashing functions to generate a ~~**MAC**~~ **Message Digest** of hashing functions to generate a ~~**MAC**~~ **Message Digest** , which is then ~~**encrypted**~~**signed** using a private RSA key. All validators of digital signatures shall be capable of using a ~~**MAC**~~ **Message Digest** generated by any of the hashing functions specified (RIPEMD-160, MD5, SHA-1 or SHA256, SHA384, SHA512).

Note: Local rules and regulations may further restrict the hashing functions that are permitted. These regulations usually restrict the hashing functions that may be used by the SCP in creating a new signature on a new SOP Instance. For example, they may prohibit use of RIPEMD-160**, and** ~~**and**~~ MD5**, and SHA-1**. The regulations usually allow an SCU to verify an old signature that uses an algorithm that is now prohibited for new signatures. Implementations that support this profile will need to accommodate these local regulations.

As a minimum, an implementation shall include the following Attributes in generating the Creator RSA Digital Signature:

a. the SOP Class and Instance UIDs

…

C.3 Authorization RSA Digital Signature Profile

The technician or physician who approves a DICOM SOP Instance for use may request the Application Entity to generate a signature using the Authorization RSA Digital Signature Profile. The Digital Signature produced serves as a lifetime data integrity check that can be used to verify that the pixel data in the SOP instance is the same that the technician or physician saw when they made the approval.

The signature shall use one of the RIPEMD-160, MD5, SHA-1 or SHA-2 family (SHA256, SHA384, SHA512),-of hashing functions to generate a ~~**MAC**~~ **Message Digest** of hashing functions to generate a ~~**MAC**~~ **Message Digest** , which is then ~~**encrypted**~~**signed** using a private RSA key. All validators of digital signatures shall be capable of using a ~~**MAC**~~ **Message Digest** generated by any of the hashing functions specified (RIPEMD-160, MD5, SHA-1 or SHA256, SHA384, SHA512).

Kommentiert [EG5]: The wording encryption digest using a private key is not correct from cryptography point of view, it is just an analogy to help people who are not familiar with cryptography to easier understand how digital signature works.
It may be accept (though not accurate) for algorithms like RSA which use private key to carry out the similar calculation as encryption during signing. However actually the signing operation does not provide confidentially protection (but non-repudiation and integrity), so it is not an encryption.
It is totally not correct for algorithm such as DSA and ECDSA where the algorithm is purely for signing.

As a minimum, an implementation shall include the following Attributes in generating the Authorization RSA Digital Signature:

a. the SOP Class and Instance UIDs

…