SMPTE ST 430-3:2012
Revision of
SMPTE 430-3-2008

SMPTE STANDARD

D-Cinema Operations — Generic Extra-Theater Message Format

Table of Contents

Approved
August 22, 2012

Foreword

SMPTE (the Society of Motion Picture and Television Engineers) is an internationally recognized standards developing organization. Headquartered and incorporated in the United States of America, SMPTE has members in over 80 countries on six continents. SMPTE's Engineering Documents, including Standards, Recommended Practices and Engineering Guidelines, are prepared by SMPTE's Technology Committees. Participation in these Committees is open to all with a bona fide interest in their work. SMPTE cooperates closely with other standards-developing organizations, including ISO, IEC and ITU.

SMPTE Engineering Documents are drafted in accordance with the rules given in Part XIII of its Administrative Practices.

SMPTE Standard 430-3 was prepared by Technology Committee 21DC.

Intellectual Property

At the time of publication no notice had been received by SMPTE claiming patent rights essential to the implementation of this Standard. However, attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. SMPTE shall not be held responsible for identifying any or all such patent rights.

1 Scope

This standard presents a specification for a generic Extra-Theatre Message (ETM) format for use with unidirectional communications channels used in security communcations for Digital Cinema (D-Cinema) systems.

The ETM specification is a generic XML security wrapper that includes specific fields which can be extended to carry different kinds of information to meet various application-level requirements. (For example, the Key Delivery Message (KDM) is a specific instance of this format.) The ETM uses W3C Extensible Markup Language (see [XML]) to represent the information payload. It provides security using the XML encryption and signature primitives.

Note: The brackets convention "[…]" as used herein denotes either a normative or informative reference.

2 Conformance Notation

Normative text is text that describes elements of the design that are indispensable or contains the conformance language keywords: "shall", "should", or "may". Informative text is text that is potentially helpful to the user, but not indispensable, and can be removed, changed, or added editorially without affecting interoperability. Informative text does not contain any conformance keywords.

All text in this document is, by default, normative, except: the Introduction, any section explicitly labeled as "Informative" or individual paragraphs that start with "Note:"

The keywords "shall" and "shall not" indicate requirements strictly to be followed in order to conform to the document and from which no deviation is permitted.

The keywords, "should" and "should not" indicate that, among several possibilities, one is recommended as particularly suitable, without mentioning or excluding others; or that a certain course of action is preferred but not necessarily required; or that (in the negative form) a certain possibility or course of action is deprecated but not prohibited.

The keywords "may" and "need not" indicate courses of action permissible within the limits of the document.

The keyword "reserved" indicates a provision that is not defined at this time, shall not be used, and may be defined in the future. The keyword "forbidden" indicates "reserved" and in addition indicates that the provision will never be defined in the future.

A conformant implementation according to this document is one that includes all mandatory provisions ("shall") and, if implemented, all recommended provisions ("should") as described. A conformant implementation need not implement optional provisions ("may") and need not implement them as described.

3 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 possibility of applying the most recent edition of the standards indicated below.

[D-Cinema Digital Certificate] SMPTE 430-2-2006, D-Cinema Operation — Digital Certificate

[FIPS-180-2] "Secure Hash Standard" Version 2. August 1, 2002. FIPS-180-2. See: http://csrc.nist.gov/publications/fips/fips180-2/fips180-2.pdf

[FIPS-197] "Advanced Encryption Standard (AES)" November 26, 2001. FIPS-197. See: http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

[FIPS-198] "The Keyed-Hash Message Authentication Code (HMAC)" March 6, 2002. File updated April 8, 2002. http://csrc.nist.gov/publications/fips/fips198/fips-198a.pdf

[PKCS1] "PKCS #1: RSA Cryptography Specifications Version 2.1" By B. Kaliski. February 2003. RFC 3447 See: http://www.ietf.org/rfc/rfc3447.txt

[RFC2253] "Lightweight Directory Access Protocol (v3):UTF-8 String Representation of Distinguished Names" December 1997. See: http://www.ietf.org/rfc/rfc2253.txt

[RFC4051] "Additional XML Security Uniform Resource Identifiers (URIs)" April 2005. See: http://www.ietf.org/rfc/rfc4051.txt

[Time] UTC, RFC 3339: Date and Time on the Internet: Timestamps. G. Klyne and C. Newman. Informational, July 2002. See: http://ietf.org/rfc/rfc3339.txt

[UUID] "A Universially Unique Identifier (UUID) URN Namespace" July 2005. See: http://www.ietf.org/rfc/rfc4122.txt

[XML] "*XML Schema Part 1: Structures"* World Wide Web Consortium May 2001. See: http://www.w3.org/TR/2001/REC-xmlschema-1-20010502

[XML-Encrypt] "XML Encryption Syntax and Processing" World Wide Web Consortium December 2002. See: http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/

[XML-Sign] "XML-Signature Syntax and Processing" World Wide Web Consortium February 2002. See: http://www.w3.org/TR/2002/REC-xmldsig-core-20020212/

4 Glossary

The following paragraphs define the acronyms used in this document.

AES: Advanced Encryption Standard secret key algorithm. Defined in [FIPS-197].

ASN.1: Abstract Syntax Notation 1.

Base64: A printable encoding of binary data. See [Base64].

FIPS: Federal Information Processing Standards of NIST.

HMAC-SHA-1: Hash-based Message Authentication Code based on SHA-1. Defined in [FIPS-198].

IETF: Internet Engineering Task Force standards group.

IP: Internet Protocol. An IETF standard.

ISO: International Standards Organization.

KDM: Key Delivery Message – An instance of the ETM. See [SMPTE 430-1]

LE: Link Encrypter.

LD: Link Decrypter.

MD: Media Decrypter.

NIST: National Institute of Standards and Technologies.

OAEP: Optimal Asymmetric Encryption Pattern. See [PKCS1].

RO: Rights Owner.

RSA: Rivest Shamir Adleman public key algorithm. Defined in [PKCS1]

SE: Security Entity. Any Digital Cinema entity that performs cryptography.

SHA-1: Secure Hash Algorithm revision 1. Defined in [FIPS-180-2].

SHA-256: Secure Hash Algoritm. Defined in [FIPS-180-2].

SM: Security Manager.

S/MIME: Secure Multipurpose Internet Mail Extensions.

SPB: Secure Processing Block.

SSL: Secure Socket Layer protocol. See [TLS].

TCP: Transmission Control Protocol. IETF standard for reliable bi-directional streams.

TLS: Transport Layer Security protocol. See [Rescorla].

TMS: Theater Management System.

X.509: A widely used and supported digital certificate standard. Refer to [D-Cinema Digital Certificate]

XML: Extensible Markup Language.

5 Overview of Generic Extra Theater Message (Informative)

Extra-Theater Messages (ETM) may be used generally between any two D-Cinema Security Entities (SE), however an ETM is particularly appropriate where a unidirectional rather than a bi-directional communications channel is employed. Such channels would be typical of those between a Distributor and an Exhibitor, or between a Studio and a Distributor. The ETM format defined in this document provides a basic message structure having a useful set of known security properties. It is intended that all D-Cinema extra-theatre messaging requirements utilize this structure in order to minimize the risk that introduction of new security messages will undermine the integrity of the security system.

The following diagram presents an overview of the generic security wrapper. The top-level XML element indicates that this structure is a D-Cinema Extra-Theatre security Message. It contains three elements (segments) for data: 1) authenticated and public (viewable by anyone who receives the message), 2) authenticated and private (viewable by the intended recipients only), and 3) authentication (signature and trust) information.

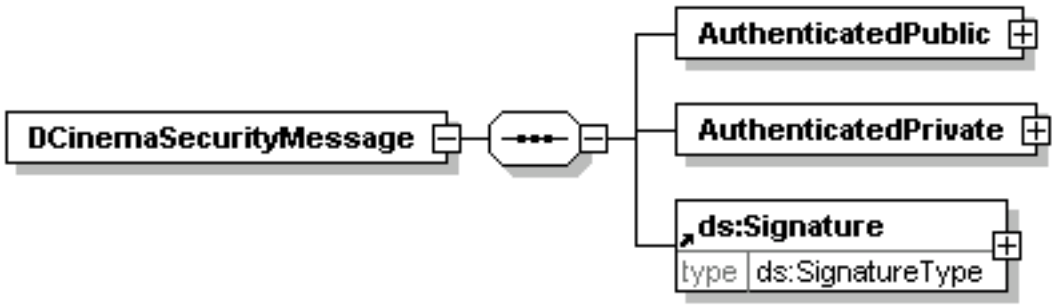


Figure 1 – XML Diagram for Generic Extra Theater Message

The AuthenticatedPublic segment includes standard message header information and a place to put required standard extension elements for the particular message type, and a place for proprietary extensions that are not critical to the baseline interoperability standard.

The single signer of the ETM is identified in the AuthenticatedPublic segment, and any entity that receives the message is able to read and authenticate the information in the AuthenticatedPublic segment. For ETMs that carry encrypted information in the AuthenticatedPrivate segment, the identity of the recipient of this private information (as specified by the issuer name and serial number of a certificate) appears in a standard field (KeyInfo) of the standard XML EncryptedKey element of the AuthenticatedPrivate segment. To avoid redundancy, the recipient information is not also carried in the AuthenticatedPublic segment.

The AuthenticatedPrivate segment includes zero or more blocks of information encrypted by RSA (called EncryptedKey) and an optional block of information encrypted by AES (called EncryptedData). The use of the EncryptedKey and EncryptedData fields is application-dependent. For example, the KDM message uses the RSA blocks in a special way and does not use the AES block. Other instances of the ETM may use the AuthenticatedPrivate segment to carry data that is hidden from all but the intended recipients. The data in this segment is encrypted with a fresh random AES key (in the EncryptedData segment), and that AES key is made available to the desired recipient in one or more EncryptedKey elements by encrypting the AES key with the public key of the recipient. The recipient has the matching private key, and so can decrypt the RSA block and recover the AES key.

The Signature segment includes 1) the value of the signer's certificate chain (note that the identity of the signer appears in the AuthenticatedPublic segment), 2) a SignedInfo segment that separately specifies the expected hash of the AuthenticatedPublic and AuthenticatedPrivate parts (this allows any entity that handles this message to detect tampering, even if it is not the intended recipient), and 3) an RSA signature on the SignedInfo element, which thus authenticates the two expected hash values that in turn authenticate the AuthenticatedPublic and AuthenticatedPrivate portions. The Signature segment is not itself authenticated, though it is believed if an attacker made any modifications to the Signature section, then the authentication of the other sections would fail.

To facilitate parsing, the ETM is represented with the UTF-8 character set. All strings intended for human display include a language attribute that is used to select an appropriate character set to display the UTF-8 string contents. All date-time values are expressed in UTC format. The cryptographic mechanisms and structures are from the XML standards for encryption and digital signatures.

6 Authenticated and Public (Unencrypted) Information

The information in this segment of the ETM shall be digitally signed, and trust in the signature can be verified using the certificate chain in the Signature portion. This segment shall not be encrypted, so any entity that has access to the message can extract this information. The word "public" that appears in the XML label for this portion means that any entity that receives the message can view this portion.

The formal XML definition is given in Annex C. Figure 2 is an informative illustration of the appropriate code section from that annex.

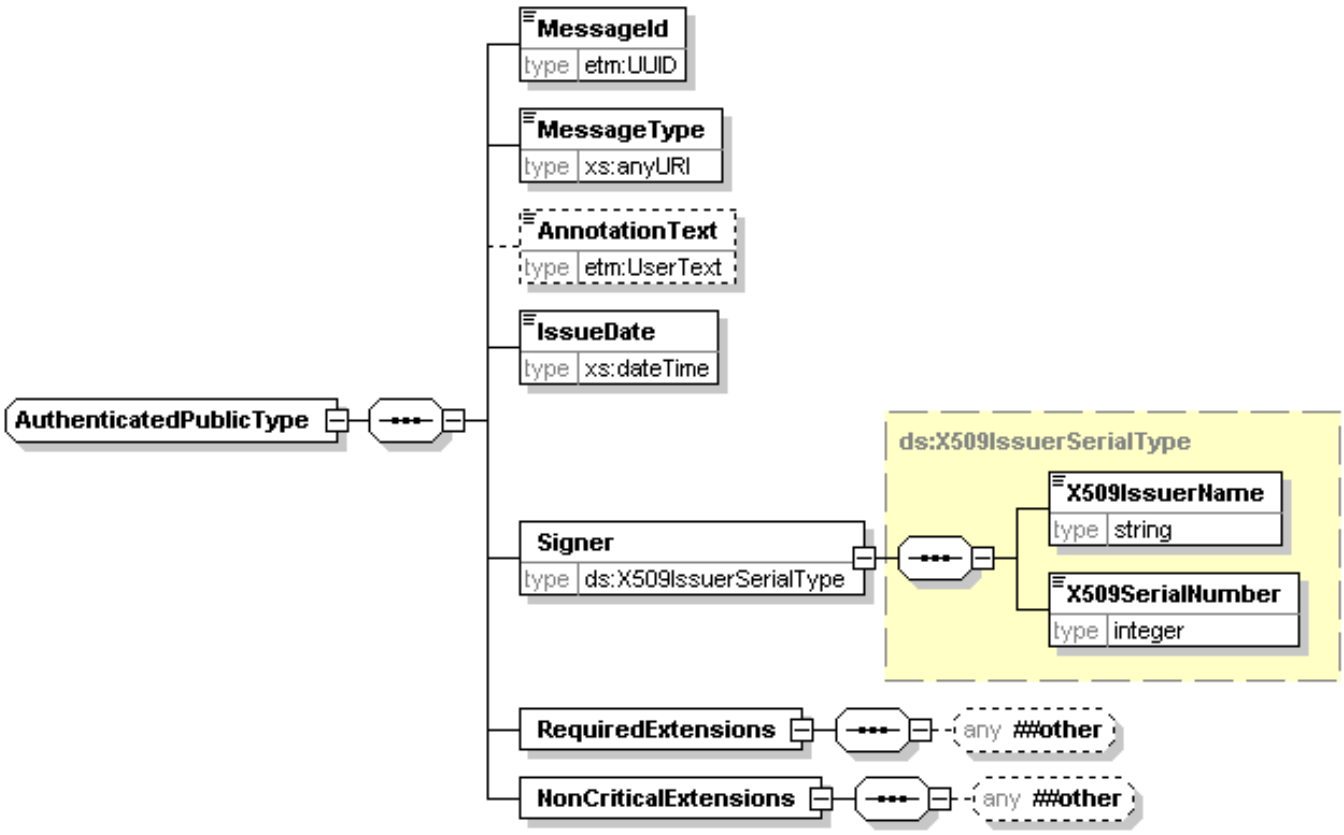


Figure 2 – Authenticated and Public Portion of ETM (Informative)

6.1 MessageId

The MessageId field shall be a globally unique identifier for a given ETM that is chosen by the creator of the message. In other words, no two otherwise different messages shall share the same Id. This value is helpful for logging, tracking and indexing ETM messages. It is in the “urn:uuid” format that is also used with the D-Cinema packing list and composition play list standards.

6.2 MessageType

The MessageType field identifies the specific version and type of the message. It is a URI string that identifies the namespace for the particular ETM instance specification being represented.

6.3 AnnotationText

The optional AnnotationText field contains a human-readable description of the message. It is not used in any security-related process and is only meant as a display hint to human users. Unless the optional xml:lang attribute is specified, the content of the field shall be en. If humans need to troubleshoot a problem related to

an ETM message, they should be able to refer to the ETM using the AnnotationText and perhaps the IssueDate. Both fields are easier for people to handle than the MessageId.

6.4 IssueDate

The IssueDate field indicates the time and date when the message was issued. The signer's certificate chain shall be valid at this time. It shall be a UTC timestamp [Time].

6.5 Signer

The Signer field identifies the certificate that may be used to validate the signature on this message. An X.509 certificate is identified by name of the Certificate Authority (CA) that issued it, called IssuerName, and the unique serial number assigned by the CA, called SerialNumber. The signer's entire certificate chain shall appear in the Signature segment of the ETM (see section 8). The Distinguished Name value in the X509IssuerName element shall be compliant with RFC 2253 [RFC2253].

6.6 RequiredExtentions (Optional)

The RequiredExtentions field shall contain zero or one opaque elements from another namespace that are required for the proper interpretation and usage of a specific ETM. It provides a place for adding information that can be visible to all entities that receive this message. The specification for each type of ETM shall provide further constraints on the structure of this field. See [KDM] as an example.

6.7 NonCriticalExtensions (Optional)

The NonCriticalExtensions field may contain zero or one opaque elements from another namespace that, depending upon design, may be optionally required for the proper interpretation and usage of a specific ETM implementation. It provides a place to carry information that is outside the scope of an associated normative interoperability specification. It shall be allowed for noncritical extensions to be ignored by receiving devices that are not capable of using or understanding the information.

7 Authenticated and Private (Encrypted) Information

This segment of the ETM shall be digitally signed, and trust in the signature can be verified using the certificate chain in the Signature portion. This portion is encrypted before being transmitted. The word "private" that appears in the XML label for this segment means that only a specified set of recipients is able to decrypt and view this information.

The formal XML definition is given in Annex C. Figure 3 is an informative illustration of the appropriate code section from that annex.

Anyone can verify the signature on the ETM and validate the certificate chain to decide whether the message has been modified and whether it was created by a trusted entity. However, only an entity that knows the private key of one of the recipients can decrypt this portion of the message.

This segment contains zero or more EncryptedKey fields and at most one EncryptedData field. The EncryptedKey field defines data (that includes an AES key) that is encrypted by the RSA algorithm and the Encrypted Data field defines data (if any) encrypted by the AES algorithm. It is a standard cryptographic security practice to use two encryption algorithms (RSA and AES) to get the key management benefit of using RSA and the performance benefits of using AES.

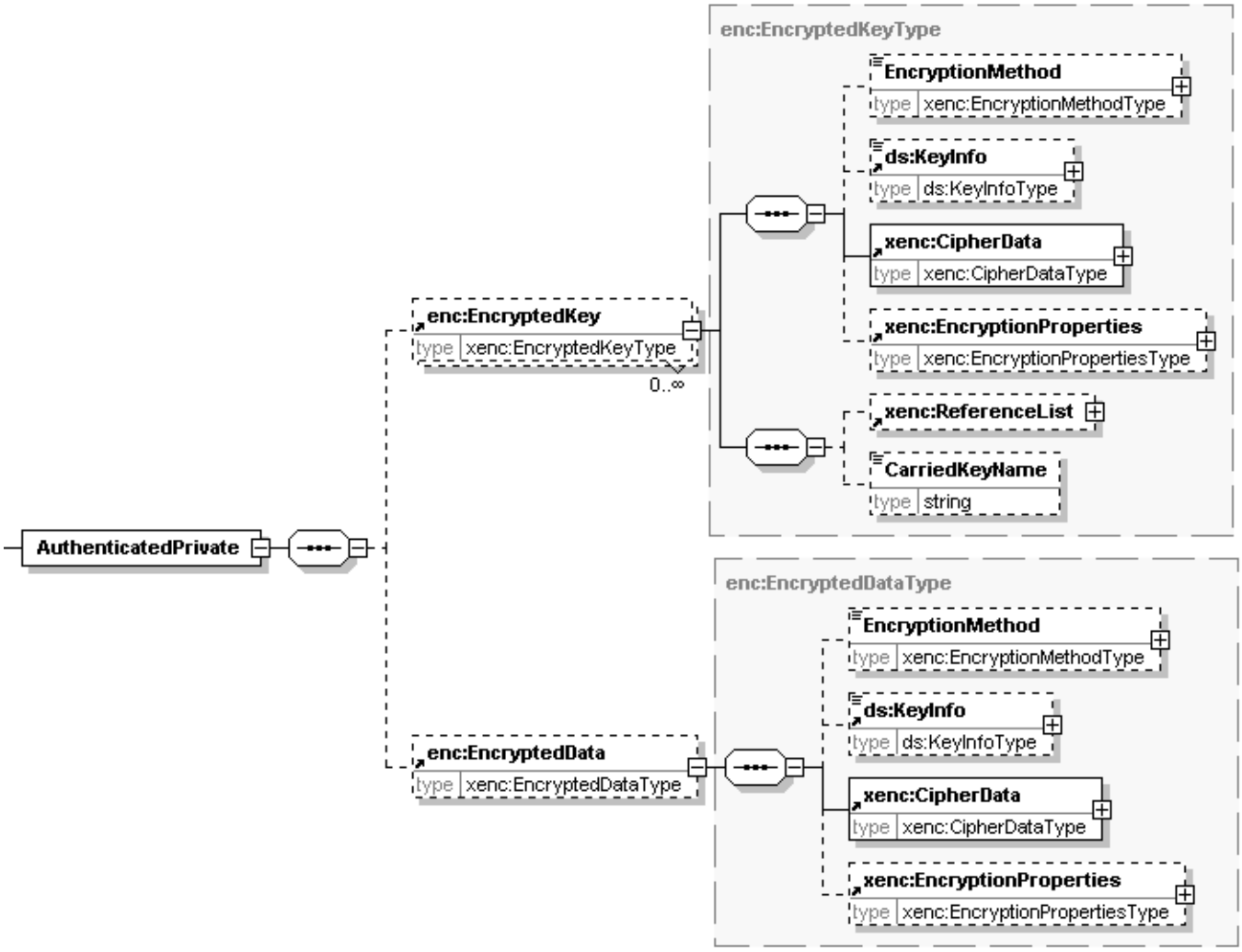


Figure 3 – Authenticated and Private Portion of ETM (Informative)

7.1 EncryptedKey

This optional element contains information encrypted with a public key algorithm, specifically RSA as defined in [PKCS1], along with all the parameters and information needed to extract that information.

7.1.1 EncryptionMethod

This field of EncryptedKey specifies the encryption algorithm and parameters. The value is a URI string that locates the algorithm specification. However, the recipient does not need to perform any network access to validate this field; just string comparison. The value of this field shall be rsa-oaep-mgf1p as shown below:

<enc:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-oaep-mgf1p"/>

All ETMs shall use the mode for RSA called Optimal Asymmetric Encryption Padding (OAEP), which is specified in [PKCS1]. The OAEP Parameter shall be omitted, causing OAEP to use a default ciphertext redundancy value. The OAEP digest algorithm shall be limited to SHA-1. The DigestMethod element shall be present and the Algorithm attribute shall be set to the URI value "***http://www.w3.org/2000/09/xmldsig#sha1***".

These choices were made to simplify the implementation and to avoid features that are not widely supported in an interoperable manner.

7.1.2 KeyInfo

This field identifies the RSA public key used to encrypt the EncryptedKey CipherData by naming the certificate that contains the public key that was used to create the ciphertext. The matching RSA private key is needed to decrypt the key. The recipient's certificate shall be named by its IssuerName and Issuer Serial Number. The format of the names and serial numbers are specified in [D-Cinema Digital Certificate].

This KeyInfo element shall only contain a single X509Data field that shall only contain an X509IssuerSerial field.

7.1.3 CipherData

This field is an RSA encrypted block of data that can be decrypted using the private key indirectly specified in the KeyInfo field. The plaintext contents of the CipherData field are application dependent. The plaintext is usually a 128-bit AES key. See [KDM] for an example of plaintext that contains more than an AES key.

The D-Cinema system uses 2048-bit RSA keys, so the ciphertext is 256-bytes long. Due to the 42-byte header that is part of the OAEP padding [see PKCS1], the plaintext can be at most 214-bytes long.

7.1.4 EncryptionProperties

This field shall not be present. The XML encryption standard includes it to provide a place to put additional information about generating or locating the RSA key or the EncryptedData field. It is omitted to simplify interoperability.

7.1.5 ReferenceList

This field shall not be present. The XML encryption standard includes it to provide a place to identify one or more EncryptedData elements that can be decrypted by the key contained in the EncryptedKey element. The ETM standard only allows at most one EncryptedData element so this information is unnecessary. It is omitted to simplify interoperability.

7.1.6 CarriedKeyName

This field is required when the EncryptedData field is included in the ETM, otherwise it shall be absent. This field is used to assign an identifying name to the AES key carried in the EncryptedKey element. The EncryptedData element, if present, has a KeyName field as part of the KeyInfo element that matches this field.

The ReferenceList and CarriedKeyName fields are both used to link the EncryptedKey element to the correct EncryptedData element. In the ETM, there can be multiple recipients for the same EncryptedData, so the EncryptedData field shall identify a key name such as "EncryptedDataKeyName" and the EncryptedKey elements for each recipient shall have a CarriedKeyName field that has the same name. This indicates that any one of the EncryptedKey elements could be decrypted to find the appropriate key.

7.2 EncryptedData (Optional)

This section describes the use of the optional EncryptedData element in ETM. A diagram of this element is shown below. This field shall not be present if there are no EncryptedKey fields.

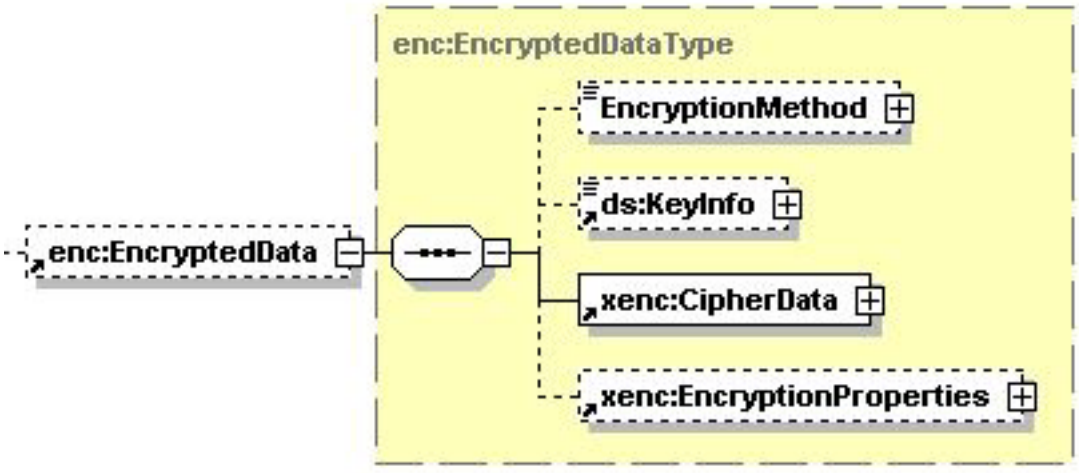


Figure 4 – EncryptedData in ETM (Informative)

The EncryptionMethod shall specify that the data be encrypted using the AES cipher with a 128-bit key operating in CBC mode. The XML Encryption Standard [XML-Encrypt] requires that the implementation generate a fresh random CBC Initialization Vector (16-bytes) and append it to the beginning of the CipherData.

The KeyName field of the KeyInfo element specifies the name of the AES key needed to decrypt the ciphertext. This name shall match the CarriedKeyName value in all of the EncryptedKey elements. The EncryptionProperties shall not be present. The CipherData shall be present. The Type attribute of the EncryptedData element shall be present and specify the XML type of the plaintext.

The XML Encryption Standard defines a new form of CBC padding that is a weaker version of PKCS#5 padding. The last byte in the last CBC block specifies the number of padding bytes (between one and 255), but the value of the remaining padding bytes is unspecified. To avoid adaptive ciphertext attacks, implementations shall treat an error with the CBC padding in the same way they treat an error with the digital signature.

8 Signature Information

This segment of the ETM provides authentication for the other sections using the primitives from the XML Digital Signature standard [XML-Sign]. Digital certificates and associated data shall use the X.509 certificate form specified for D-Cinema in [D-Cinema Digital Certificate]. The following diagram illustrates the Signature segment.

The Signature primitive that is defined in the XML Digital Signature standard is quite flexible. For all ETM messages, it shall be used as specified in this section.

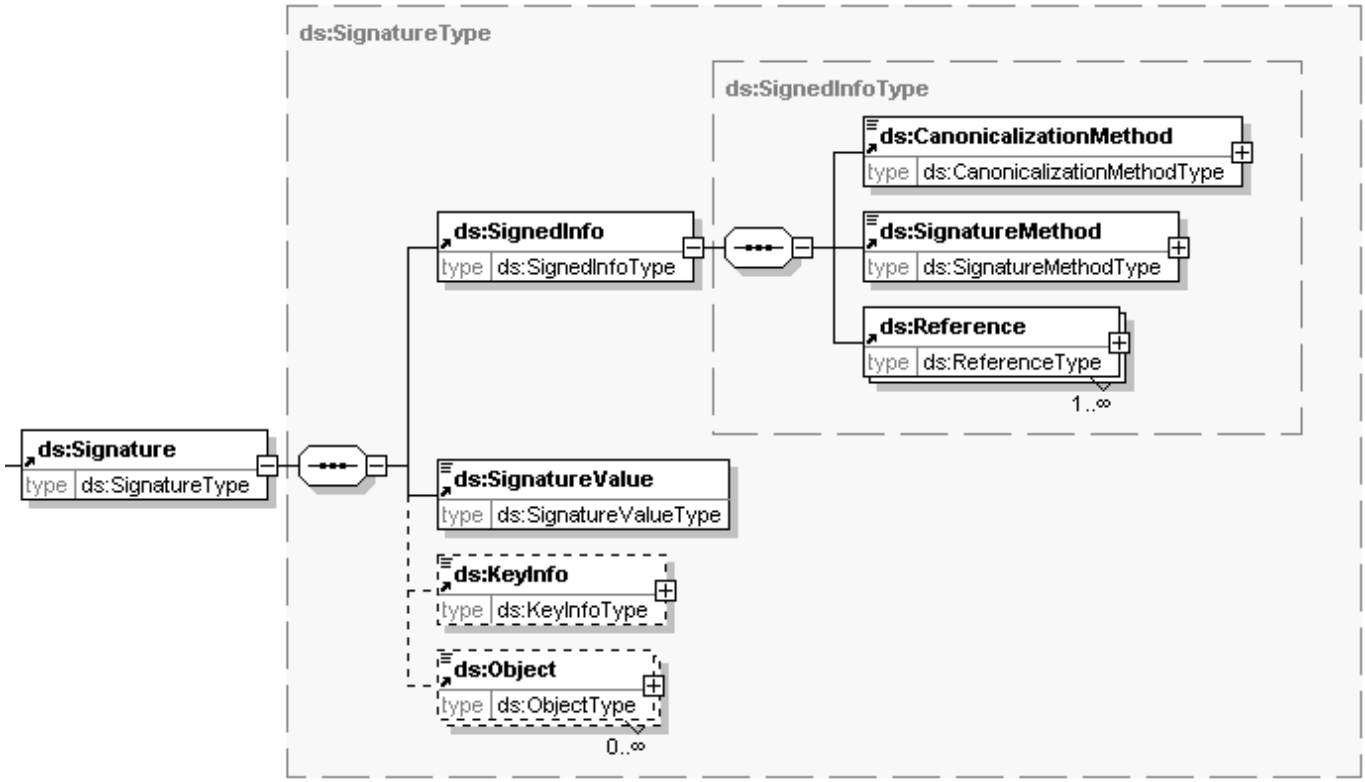


Figure 5 – Signature Section of ETM (Informative)

8.1 XML Embedding

The ability to validate and interpret a fragment of XML depends on its surrounding XML context. Use of namespace attributes in XML documents creates an unambiguous context at the document level. However when an XML fragment smaller than a document is reused within another document, there is potential for misinterpretation or validation failure. The D-Cinema application calls for embedding signed data structures, such as CPLs, in other XML documents, such as intra-theatre messages. This raises difficulties that have been encountered by other XML-based standards projects (e.g. the effort to embed security assertions into SOAP messages). In particular, the required canonicalization algorithm REC-xml-c14n-20010315 is liable to insert namespace declarations within a data structure when it is being canonicalized; this shall cause signature verification to fail.

The following approach shall be used by D-Cinema applications.

Each signed data structure to be embedded (e.g. CPL) should have a “native” or “original” form in which it is a complete document. The contents of the document header (prolog) shall be completely specified by the controlling standards document. During the embedding process only the prolog is stripped and the remainder of the document is embedded intact. As a universal rule, the semantics of the embedded fragment at the recipient shall be governed by the result of “de-embedding” – i.e. reconstructing the original document – which is achieved by re-attaching the standard prolog to the embedded content. This de-embedding shall be done before any canonicalization, and therefore no namespace interactions are possible.

This corresponds to option 1 in xmldsig-core Section 8.3: “Rely upon the enveloping application to properly divorce its body (the signature payload) from the context (the envelope) before the signature is validated” [see XML-Sign].

Informative Note: Implementors should note that the base64Binary datatype as defined in "XML Schema Part 2: Datatypes, W3C Recommendation, May 2001", as referenced by the normative reference "XML-Signature Syntax and Processing, W3C Recommendation, February 2002", constrains its content (i.e. the Base64-encoded data) to 76 characters per line.

8.2 SignedInfo

The XML Digital Signature standard [see XML-Sign] defines a two-step process for checking signatures. First the actual hash values of different portions of the ETM are computed and compared against the expected values that appear in the Reference elements of the SignedInfo. Next, the SignedInfo element is canonicalized and then hashed and finally verified against the SignatureValue. The expected hash values for different portions of the ETM are listed in the Reference elements of the SignedInfo along with the identification (a document-relative URI) of the portion of the message for which it is the hash value.

Informative Note: Implementations based on security chips may reverse the two steps. First, the chip could validate the SignedInfo element and record the hash values that appear in that element in some form of high integrity storage, and second, the different portions of the message could be passed to the chip, which would compare the computed hash values to the expected hash values.

The CanonicaliazationMethod field shall be REC-xml-c14n-20010315#WithComments, which includes any comments that appear in the SignedInfo element. This prevents attackers from adding misleading comments. The SignatureMethod field shall be set to the URI value “http://www.w3.org/ 2001/04/xmldsig-more#rsa-sha256” [RFC 4051]. This specifies SHA-256 as the hash algorithm and RSA as the signature algorithm.

For all ETMs, the SignedInfo shall contain at least two Reference fields. The first is the hash of the AuthenticatedPublic element and the second is the hash of the AuthenticatedPrivate element (after encryption has been performed). There may be a third Reference field that specifies the hash value for the plaintext decrypted from the EncryptedData element. If the AuthenticatedPrivate node is empty (e.g., <AuthenticatedPriavate Id=” AuthenticatedPrivateLabel” />), then there shall still be a reference to and a hash value for that node.

The URI attribute of the Reference elements shall identify the nodes being hashed. This URI is a document relative pointer label that matches the Id attribute of the AuthenticatedPublic node, the AuthenticatedPrivate node, or other nodes.

Note that the interpretation of the URI identifier shall be based on treating the ETM as a separate XML document. This avoids potential identifier conflicts that could arise if several ETM messages are embedded into a larger XML document. For example, one allowable implementation is to use the same label (e.g., Id=” AuthenticatedPrivateLabel”) for all AuthenticatedPrivate nodes for all ETM generated by that implementation. If two such ETMs were embedded in the same larger XML document, then the labels would be duplicates. The resolution is to treat each ETM as a separate XML document before performing a signature checks.

The Id and Type attributes of the Reference elements are optional. A conforming implementation is allowed to ignore these values.

The Transforms field of the Reference elements shall be omitted, meaning the bytes of the referenced node are hashed without any transformations. To ensure that the signer and verifier compute the hash of the same bytes, the transportation mechanisms shall not modify the bytes of this XML document (e.g., the document should not be converted to another character set or CRLF representation).

The DigestMethod field of all the Reference elements shall be http://www.w3.org/2001/04/xmlenc#sha256.

The DigestValue field of all the Reference elements shall be the Base64 encoded output of the SHA-256 hash of the referenced node.

8.3 SignatureValue

The SignatureValue element shall be the output of the http://www.w3.org/2001/04/xmldsig-more#rsa-sha256 operation that is used to generate the signature.

8.4 KeyInfo Certificate Chain

The signer's certificate shall be identified by its IssuerName and SerialNumber in the Signer element in the AuthenticatedPublic segment of the message. It shall not be similarly (specifically) identified in this KeyInfo element.

The entire certificate chain of the signer, including the root certificate, shall be carried in the KeyInfo element as a sequence of X509Data elements. Each of the X509Data elements shall correspond to one certificate in the chain, and contain one X509IssuerSerial element and one X509Certificate element. The certificates may appear in any order.

8.5 Object Information

The Object field of the Signature element shall not be present.

Annex A Design Features and Security Goals (Informative)

Design features of the ETM

- The type and version of the message is clearly identified using a unique XML namespace.

- Each message has a globally unique identifier that can be used, for instance, to index the message for storage or target the message for revocation or duplicate detection.

- The message includes a NonCriticalExtensions element to support vendor-specific extensions. As the name of the element implies, these extensions are not necessary to process the message and may be ignored by baseline interoperable implementations.

- The message has exactly one signer and one or more recipients. The identity of the signer and the recipients is part of the information that is signed. This prevents a message intended for one recipient from being modified to resemble a message to another recipient. The signer knows and expresses the identity of the recipients and the issuance time of the message when the signature is created. The issuance time may differ from the signature time, enabling the signer to pre-sign a collection of messages that shall all be issued at the same time.

- The message has three segments. One segment is authenticated and public (anyone can see the information and anyone can verify the signature of this information). A second segment is authenticated and private (anyone can verify that the encrypted information has not been modified, though only the intended recipient can decrypt this part of the message). The third segment contains signature information including the signer's complete certificate chains, which avoid the need to create any out-of-band mechanism for distributing certificates.

- The XML scheme for the message is valid in both encrypted and unencrypted forms, so standard XML cryptography toolkits and XML parsing tools can be used throughout the handling of the ETM.

- Element names are chosen to be unique in order to enable XML parsers that are implemented in a small amount of memory. For example the elements MessageId and KeyId are both UUID values, and could be just called "Id", but then the XML parser would have to keep track of nesting information to locate the correct one.

- The EncryptedData element has a parameter that defines the type of the plaintext element it protects. ETM recipients shall check that the decrypted element matches this type. This helps prevent attacks based on slicing the EncryptedData originally carried by one kind of message into the body of another kind of message.

- The AuthenticatedPublic element has a parameter that defines the type of message that it is part of. The signature protects the integrity of this parameter as well as the integrity of the fields inside that element. This helps prevent attacks based on slicing the AuthenticatedPublic element originally carried by one kind of message into the body of another kind of message.

- The EncryptedKey element in the recipient's element is encrypted with the OAEP padding to avoid various adaptive chosen ciphertext attacks (e.g., the attacker sends modified ciphertext to the recipient until certain error messages are returned that provide valuable information to the attacker).

- Consideration was given to using enveloped-signature for the whole ETM rather than the three-hash scheme. (distinct hashes for authenticated public, private and decrypted auth-private elements). This would simplify cryptographic processing at the cost of creating some subtle security risks. The security principle is that the plaintext of a message must be signed as well as the ciphertext; this ensures that the signer knows what he is signing. The more subtle cryptographic attack has to do with the ability of an attacker to splice together ciphertext that resulted from two different signed

messages. If only the ciphertext is signed, this attack can succeed with probability 1 in 2**64 (for AES, which has 128bit blocks) instead of 1 in 2**128 for signatures based on the 256-bit SHA-256 hash. This argues against using enveloped-signature.

- Another problem with using enveloped-signature is that the list of certificates included in the message are not covered by the signature. Most importantly the signer's certificate is not included. Given that the system allows multiple root certificates, it is very important that the certificate of the signer be included in the data validated by the signer. If the signer is just identified by issuer-name and serial number, an attack using a different certificate authority might be able to create a fake certificate. By using the given signing method, the public key of the signer, which appears in the signer's certificate, is bound to the message, and thus these kinds of attacks are thwarted.

Security goals of the ETM

This section describes the security goals of the ETM message. These goals describe the security benefits and threat resistance features that the ETM is believed to achieve. Specific extensions to the ETM, for example those in the KDM, can be made to achieve additional goals.

- Resists man-in-the-middle attack.
- Resists replay attack.
- Resists turn-around attacks (making sender of a message think that he is the intended recipient of the message).
- Resists message splicing attack (combining parts of one or more valid messages to create a new valid message).
- Resists AES-CBC truncation attacks.
- Resists AES-CBC splicing attacks.
- Resists protocol versioning attacks (forcing parties to use an earlier version of the protocol).
- Resists protocol algebra attacks (using features of one message in the system to defeat security goals of another message in the system).
- Resists cross-system attacks (the security of this system is not compromised if the same keys are used with another system; for example, an RSA private key could be used to receive KDM messages and to set up TLS sessions).
- Detects tampering with a 256-bit cryptographic checksum.
- Resists adaptive ciphertext attack on RSA envelope.
- System architecture ensures graceful degradation of security. If an attacker has compromised a few keys or has short-term access to keys or certificate issuing systems, the architecture looses only some of its security properties.
- System architecture ensures reasonable recovery from expected compromises.
- Signer of message sees plaintext of message.
- Signature covers the list of recipients (a “you are fired” message sent from Alice to Bob cannot be made to look like a message from Alice to Carol).
- Non-recipients can validate that the message has not been tampered with.

Annex B Bibliography (Informative)

This section contains informative references that provide helpful background information.

[ASN.1] For a collection of useful links to ASN.1 resources see:
http://www.cs.columbia.edu/~hgs/internet/asn.1.html

[Base64] MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for Specifying and Describing the Format of Internet Message Bodies. See: http://www.ietf.org/rfc/rfc1521.txt

[Gutmann] "X.509 Style Guide" By Peter Gutmann. See:
http://www.cs.auckland.ac.nz/~pgut001/pubs/x509guide.txt

[KDM] SMPTE 430-1-2006, D-Cinema Operations — Key Delivery Message

[NIST-KMG] "Key Management Guideline" Draft of June 3, 2002. NIST. See:
http://csrc.nist.gov/encryption/kms/guideline-1.pdf

[Rescorla] Eric Rescorla. SSL and TLS: Designing and Building Secure Systems. Addison Wesley Professional. ISBN 0201615983. October 2000.

[RFC3280] "Internet X.509 Public Key Infrastructure Certificate and CRL Profile" by R. Housley, W. Ford, W. Polk, D. Solo, April 2002. See: http://www.ietf.org/rfc/rfc3280.txt

[RFC2693] "SPKI Certificate Theory" by C. Ellison, B. Frantz, B. Lampson, R. Rivest, B. Thomas, and T. Ylonen, September 1999. See: http://www.ietf.org/rfc/rfc2693.txt

[RFC4055] "Additional Algorithms and Identifiers for RSA Cryptography for Use in the Internet X.509 Public Key Infrastructure" by J. Schaad, B. Kaliski, R. Housley, June 2005. See: http://www.ietf.org/rfc/rfc4055.txt

[TLS] "The TLS Protocol Version 1.0." by T. Dierks and C. Allen. January 1999. IETF RFC 2246. See:
http://www.ietf.org/rfc/rfc2246.txt

[X.509] ITU-T Recommendation X.509 (1997 E): Information Technology - Open Systems Interconnection – The Directory: Authentication Framework, June 1997.

[XML_KMS] " XML Key Management Specification (XKMS)" World Wide Web Consortium Draft April 2003. See: http://www.w3.org/TR/xkms2/

[ISO 3166] Codes for the Representation of Names of Countries (ISO 3166-1993 (E)) See:
http://www.iso.org/iso/en/prods-services/iso3166ma/index.html

Annex C XML Schema for ETM (Normative)

The XML Schema document presented in this appendix normatively defines the structure of an Extra Theater Message using a machine-readable language. While this schema is intended to faithfully represent the structure presented in the normative prose portions (Sections 5 through 8) of this document, conflicts in definition may occur. In the event of such a conflict, the normative prose shall be the authoritative expression of the standard.

Note: XML Schemas for the “EncryptedData” element and “Signature” segment are defined above at the end of Sections 7.2 and 8 respectively.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- 2006-03-03-jhurst -->
<xs:schema
  targetNamespace="http://www.smpte-ra.org/schemas/430-3/2006/ETM"
  xmlns:etm="http://www.smpte-ra.org/schemas/430-3/2006/ETM"
  xmlns:enc="http://www.w3.org/2001/04/xmlenc#"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:import namespace="http://www.w3.org/2000/09/xmldsig#"
schemaLocation="http://www.w3.org/TR/2002/REC-xmldsig-core-
20020212/xmldsig-core-schema.xsd"/>
  <xs:import namespace="http://www.w3.org/2001/04/xmlenc#"
schemaLocation="http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/xenc-
schema.xsd"/>
  <xs:element name="DCinemaSecurityMessage"
type="etm:DCinemaSecurityMessageType"/>
  <xs:complexType name="DCinemaSecurityMessageType">
    <xs:sequence>
      <xs:element name="AuthenticatedPublic"
type="etm:AuthenticatedPublicType"/>
      <xs:element name="AuthenticatedPrivate"
type="etm:AuthenticatedPrivateType"/>
      <xs:element ref="ds:Signature"/>
    </xs:sequence>
  </xs:complexType>

// The code below is described in Section 6. //

  <xs:complexType name="AuthenticatedPublicType">
    <xs:sequence>
      <xs:element name="MessageId" type="etm:UUID"/>
      <xs:element name="MessageType" type="xs:anyURI"/>
      <xs:element name="AnnotationText" type="etm:UserText"
minOccurs="0"/>
      <xs:element name="IssueDate" type="xs:dateTime"/>
      <xs:element name="Signer" type="ds:X509IssuerSerialType"/>
      <xs:element name="RequiredExtensions">
        <xs:complexType>
          <xs:sequence>
```

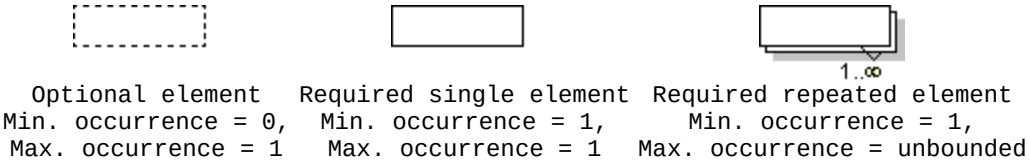
```
            <xs:any namespace="##other" processContents="strict"
minOccurs="0"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="NonCriticalExtensions">
        <xs:complexType>
          <xs:sequence>
            <xs:any namespace="##other" processContents="strict"
minOccurs="0"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
    <xs:attribute name="Id" type="xs:ID" use="required"/>
  </xs:complexType>

     // The code below is described in Section 7. //

  <xs:complexType name="AuthenticatedPrivateType">
    <xs:sequence>
      <xs:element ref="enc:EncryptedKey" minOccurs="0"
maxOccurs="unbounded"/>
      <xs:element ref="enc:EncryptedData" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="Id" type="xs:ID" use="optional"/>
  </xs:complexType>
  <xs:simpleType name="UUID">
    <xs:restriction base="xs:anyURI">
      <xs:pattern value="urn:uuid:[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-
F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{12}"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:complexType name="UserText">
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="language" type="xs:language" use="optional"
default="en"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:schema>
```

Annex D XML Diagram Legend (Informative)

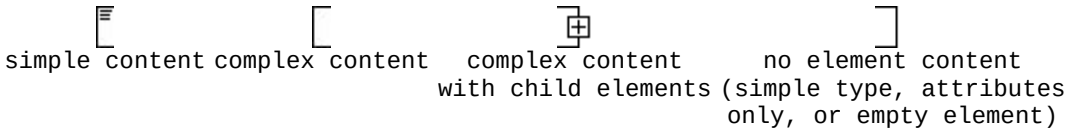
The following provides a legend for notation used in diagrams depicting XML structures.

Element symbols

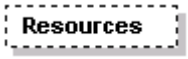
In the schema design diagrams presented above in this document, only the elements are drawn. Attributes are not visible. The cardinality of the element (0..1, 1 exactly, 0..n, 1..n) is indicated by the border of the elements. Optional elements are drawn with a dashed line, required elements with a solid line. A maximum occurrence greater one is indicated by a double border.



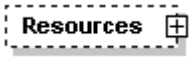
The content model of elements is symbolized on the left and right side of the element boxes. The left side indicates whether the element contains a simple type (text, numbers, dates, etc.) or a complex type (further elements). The right side of the element symbol indicates whether it contains child elements or not:



Examples



Optional single element without child elements. Minimum Occurrence = 0, Maximum Occurrence = 1, content = complex.



As above, but with child elements. he "plus" at the right side indicates the presence of one or more undisplayed child elements.

Mandatory single element. Minimum Occurrence = 1, Maximum Occurrence = 1, content = complex, no child elements (i.e. this denotes an *empty element*). The gray or green text below the element displays the xml-schema annotation associated with the element.

Mandatory multiple element containing child elements (content = complex). This element must occur at least once (Minimum Occurrence = 1) and may occur as often as desired (Maximum Occurrence = unbounded).

Mandatory single element with containing simple content (e.g. text) or mixed complex content (e.g. text with xhtml markup). Minimum Occurrence = 1, Maximum Occurrence = 1, type = xsd:string (for example), content = simple. The three lines in the upper left corner are used for both text and numeric content.

Model symbols ("compositors")

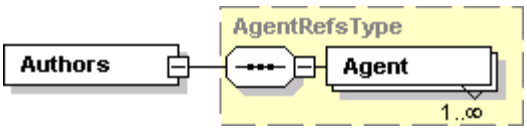
A sequence of elements. The elements must appear exactly in the sequence in which they appear in the schema diagram.

A choice of elements. Only a single element from those in the choice may appear at this position.

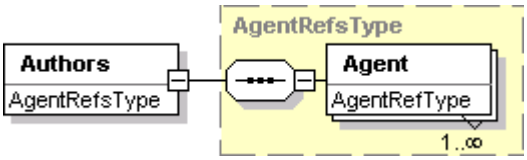
The "all" model, in which the sequence of elements is not fixed.

Types

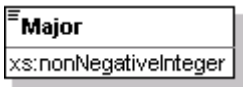
If an element refers to a complex global type, the type is shown with a border and yellow background. You can click on the gray type name shown at the top to jump to the type definition itself.



Depending on the settings in xml spy when generating the schema diagrams, the type name may be shown in the line below the element name:



In that case, the type names of simple types are shown as well:



Model groups and references

An *element group* is a named container with one or several elements. The group of elements can be reused at multiple places in the schema. Model groups are invisible in the instance document (in contrast to types, which require). Model groups have been used sparingly since they do not map to a feature in object-oriented programming languages (unless they support multiple inheritance).

Import note on reading the diagrams for model groups: If the model group symbol is drawn with simple lines (i.e. not dashed), this does not imply that the elements in the model group are required. The optionality of the group depends on the optionality of elements contained in the model group. (Model groups can be made optional; e.g., to make a model group with required elements optional in some cases, but this has not been used.)

The *"any"* group is a special kind of model group. It is a placeholder for elements not defined in the schema. The "any" element defines points where the schema can be extended. After the "Any" keyword the namespace from which the elements may come is defined, for example, "##other" specifies that the extension elements may come from any namespace, except from the current schema namespace.

Element references are indicated through a link arrow in the lower left corner. They are similar to references to model groups within a schema, but instead of refining the model group, they directly refer to a single global element. The global element can then be reused in multiple places.

Revision Notes

This version incorporates Amendment #2 to SMPTE ST 430-3 approved August 22, 2012. The changes are summarized below.

1. The Intellectual Property section has been added.

2. The last paragraph in Section 7.1.1 EncryptionMethod has been revised.