SMPTE ST 430-2:2017
Revision of
SMPTE 430-2-2006

SMPTE STANDARD

D-Cinema Operations — Digital Certificate

Table of Contents

Page

Approved
January 12, 2017

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 its Standards Operations Manual.

SMPTE ST 430-2 was prepared by Technology Committee 21DC.

Intellectual Property

SMPTE draws attention to the fact that it is claimed that compliance with this Standard may involve the use of one or more patents or other intellectual property rights (collectively, "IPR"). The Society takes no position concerning the evidence, validity, or scope of this IPR.

Each holder of claimed IPR has assured the Society that it is willing to License all IPR it owns, and any third party IPR it has the right to sublicense, that is essential to the implementation of this Standard to those (Members and non-Members alike) desiring to implement this Standard under reasonable terms and conditions, demonstrably free of discrimination. Each holder of claimed IPR has filed a statement to such effect with SMPTE. Information may be obtained from the Director, Standards & Engineering at SMPTE Headquarters.

Attention is also drawn to the possibility that elements of this Standard may be subject to IPR other than those identified above. The Society shall not be responsible for identifying any or all such IPR.

Introduction

This standard presents a specification for Digital Certificates. This standard defines the Digital Certificate format and associated processing rules in sufficient detail to enable vendors to develop and rollout interoperable security solutions.

This Digital Certificate standard is based on a constrained form of the X.509v3 format and processing rules. X.509v3 certificates have been widely used in other well-respected security standards such as SSL/TLS secure internet access, IPSec Virtual Private Networks and S/MIME secure email. The specific constraints on the X.509v3 format are chosen to reduce the amount of time and implementation effort required to achieve interoperability with high security and yet provide a robust flexible foundation that can support future enhancements. These certificates support a simple yet flexible trust model without having to introduce new business entities. Specifically, there is no need to create an industry wide certification lab, though one could be supported.

These certificates are used in several D-Cinema standards. They are used to provide authenticity and integrity for Composition Play Lists [CPL] and Packing Lists [PL]. They provide authenticity, integrity and confidentiality in Extra-Theatre Messages [ETM] such as the Key Delivery Message [KDM], and they are used with the TLS session security protocol to protect Intra-Theater Messages.

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

1 Scope

This standard presents a specification for Digital Certificates. The standard defines the Digital Certificate format and associated processing rules in sufficient detail to enable vendors to develop and implement interoperable security solutions.

The Digital Certificate standard is based on a constrained form of the X.509v3 [X.509] format and processing rules. Only the most widely supported features of X.509v3 are used in order to give vendors a large selection of X.509v3 development toolkits and certificate issuing products. The constraints also avoid the complexity and ambiguity that often occurs in systems that use X.509v3 certificates.

In the D-Cinema environment, certificates have these primary applications:

- Establishing identity of security devices
- Supporting secure communications at the network layer (e.g. TLS) or application-messaging layer (e.g., Extra Theater Messages [ETM])
- Authentication and integrity requirements for Composition Play Lists (CPL) and Packing Lists (PL)

2 Normative References

The following standards contain provisions which, 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.

[ASN.1] ISO/IEC 8824-1:2002 (ITU-T X.680, Information Technology) - Abstract Syntax Notation One (ASN.1). See: http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=35684

[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

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

[PKCS1] “PKCS #1: RSA Encryption Version 2.1” By B. Kaliski. February 2003. IETF RFC 3447. See: http://www.ietf.org/rfc/rfc3447.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

[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

[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

[X.509] ITU-T Recommendation X.509 (1997 E): Information Technology — Open Systems Interconnection — The Directory: Authentication Framework, June 1997. See: http://www.itu.int/ITU-T/asn1/database/itu-t/x/x509/1997/

3 Glossary

The following paragraphs define the acronyms used in this standard.

ASN.1: Abstract Syntax Notation 1.

BER: Basic Encoding Rules for ASN.1 structures. There are multiple BER encodings for a given value.

Base64: A printable encoding of binary data. Defined in [Base64].

CA: Certificate (issuing) Authority

DC: Digital Cinema.

DER: Distinguished Encoding Rules for ASN.1 structures. These rules create a canonical representation.

ETM: Extra Theatre Message.

FIPS: Federal Information Processing Standards of NIST.

IETF: Internet Engineering Task Force standards group.

IP: Internet Protocol. An IETF standard.

ISO: International Standards Organization.

LE: Link Encryptor.

LD: Link Decryptor.

MD: Media Decryptor.

NIST: National Institute of Standards and Technologies.

RO: Rights Owner.

RSA: Rivest Shamir Adleman public key algorithm.

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

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

SHA-256: Secure Hash Algorithm. See [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: Theatre Management System.

X.509: A widely used and supported digital certificate standard.

XML: Extensible Mark-up Language.

4 Overview of Digital Certificates (Informative)

Digital certificates provide a way for a device to start with a small amount of trustworthy information and use that to verify the trustworthiness of additional information. Certificates also support the privacy, integrity and authenticity of communications.

The certificate for a security device is a statement signed by the vendor of the device saying “If you speak to an entity that can prove that it has current access to the private key that matches the public key in this certificate, then I, the vendor of the device, state that the entity has the following attributes.” The body of the certificate lists attributes such as the make, model and serial number of the device, and the roles supported by the device.

For reasons of scaling and security, equipment vendors need not directly sign the certificates of devices. Instead there may be one or more intermediate certificates in a chain. The vendor’s primary certificate is the “root” of this chain (called the root certificate), and the device’s certificate is the “leaf-end” of the chain. The public key in the vendor’s root certificate (which is self-signed) may be used to verify the attributes in an

intermediate certificate. Those attributes include the public key of the intermediate Certificate issuing Authority (CA), which is then used to verify the next certificate in the chain, and so forth. Eventually, the public key from the last CA certificate in the chain is used to verify the device's certificate, and thus establish the trustworthiness of the attributes in the certificate (including the device's public key).

Devices that perform certificate chain validation assume that the vendor has established good policies and procedures for securely operating the CAs in the chain, which should make it unlikely that an attacker will be able to create fraudulent certificates. The name of the organization that owns the root certificate appears in all the certificates in the chain and this serves as an indication of the quality of the policies and procedures.

5 Certificate Fields

Digital certificates conforming to this specification shall use the standard X.509 (version 3) (see [X.509]) format in constrained ways defined in this standard in order to reduce the complexity and ambiguity that often occurs in systems that used X.509 certificates. This section defines those constraints.

5.1 Required Fields

This section specifies the required fields in certificates. The following table summarizes the required fields. Table 1 describes the detailed constraints for each field. The certificate shall be encoded (converted to bytes) using the ASN.1 DER rules (see [ASN.1], [Kaliski]), which produce a unique representation for the certificate.

Table 1 – Required X.509v3 fields for Digital Certificates

Field	Description
The first two fields shall appear outside of the signed portion of the certificate.	
SignatureAlgorithm	Identifier of the algorithm used to sign this certificate. Must be same as signature field inside the certificate.
SignatureValue	Value of the signature for the certificate.
The following fields are inside the signed portion of the certificate. The fields after the SubjectPublicKeyInfo field shall appear in the "extensions" part of the signed portion.	
Version	Indicates X.509 Version 3 format certificates.
SerialNumber	Serial number of certificate that is uniquely chosen by the Issuer.
Signature	Identifier of the signature algorithm. It appears inside the signed portion of the certificate and must match the algorithm identified on the outside in the SignatureAlgorithm field.
Issuer	Name of entity that issued and signed this certificate.
Subject	Name of the entity that is the subject of this certificate and thus controls access to the private key that corresponds to the public key that appears in this certificate.
Validity	Date/Time range when the certificate is valid.
SubjectPublicKeyInfo	Information about the subject's public key including the algorithm type, any algorithm parameters and the set of values that makes up the public key, such as modulus and public exponent for RSA.
AuthorityKeyIdentifier	This field identifies the issuer's certificate.
KeyUsage	Collection of flag bits that identify all the operations that are authorized to be performed with the public key in this certificate, and thus imply what can be done with the corresponding private key.
BasicConstraint	This field indicates whether certificate signing is allowed and specifies the maximum number of certificate signing certificates that can appear in the chain below this one.

Digital certificates may contain other extension fields that are meaningful to equipment from specific vendors. All implementations shall ignore extensions (i.e. fields other than the above specified required fields) that they do not understand.

5.2 Field Constraints

Table 2 describes the constraints on the required fields.

Table 2 – Field Constraints for Digital Certificates

X.509 Field	Description
SignatureAlgorithm	Shall be sha256WithRSAEncryption, which is the algorithm identifier for encrypting a SHA-256 (see [FIPS-180-2]) digest of the certificate body with RSA using PKCS #1 v1.5 signature padding (see [PKCS1]).
SignatureValue	This field is an ASN.1 Bit String that contains a PKCS #1 signature block. It shall contain a SHA-256WithRSA signature (see [RFC4055]).
Version	Shall indicate X.509 Version 3 format certificates.
SerialNumber	Unique number assigned by Issuer. Shall be an unsigned integer value that is 64-bits in length or less.
Signature	Shall be sha-256WithRSAEncryption algorithm identifier.
Issuer	Globally unique name of entity that issued and signed this certificate. See the section on Naming and Roles, for further constraints.
Subject	Globally unique name of the entity that controls access to the private key that corresponds to the public key of this certificate. See the section on Naming and Roles, for further constraints.
Validity	The issuer shall always encode certificate validity dates through the year 2049 as UTCTime (two digit years); certificate validity dates in 2050 or later shall be encoded as GeneralizedTime (four digit years). ([Time])
SubjectPublicKeyInfo	This shall describe an RSA public key. The RSA public modulus shall be 2048-bits long. The public exponent shall be 65537. The same public key may appear in multiple certificates. Certificate issuers should try to ensure that when a public key appears in multiple certificates, those certificates correspond to the same entity or device.
AuthorityKeyIdentifier AuthorityCertIssuer AuthorityCertSerialNumber	Shall be present in all certificates, including root certificates. These attributes are the unique identifier for the issuer's certificate. They name the issuer of the issuer's certificate and the serial number assigned by the issuer's issuer.
KeyUsage	Shall be present in all certificates, including root certificates. For certificate signing certificates, only the KeyCertSign flag shall be true. For leaf certificates either or both of the DigitalSignature and KeyEncipherment flags shall be true. Other flags may be true.
BasicConstraint	This field shall be present in all certificates. When present, the CA attribute shall be true only for certificate signing certificates. Otherwise the CA attribute shall be false, and the PathLenConstraint shall be absent (or zero). See example in Section 6.2.5.

5.3 Naming and Roles

This section defines the semantics of the attributes that appear in the Issuer name field and the Subject name field of certificates.

Each entity that is the subject or issuer of acertificate is unambiguously identified by a number of attributes. In order to enable the mapping of these attributes into the X.509 name structure, this specification overloads the semantics of the X.509 name attributes, as summarized in Table 3. Overloading was chosen rather than defining new attribute types in order to facilitate implementation with widely available services and toolkits.

Table 3 – Mapping of Identity Attributes to X.509 Name Attributes

Identity Attribute	X.509 Name Attribute	Description
Public Key Thumbprint	dnQualifier	Unique thumbprint of the public key of the entity issuing the certificate or being issued the certificate.
n/a	CountryName	This X.509 name attribute shall not appear in certificates conforming to this specification.
Root Name	OrganizationName	Name of the organization holding the root of the certificate chain.
Organization Name	OrganizationUnitName	Name of the organization to which the issuer or subject of the certificate belongs. This field does not identify the end owner or facility; rather it identifies the device maker.
Entity Name	CommonName	Entity issuing the certificate or being issued the certificate. See Entity Name and Roles section.

5.3.1 Public Key Thumbprint (DnQualifier)

Exactly one instance of the DnQualifier attribute shall be present in the Subject name and the Issuer name. It is a Base64 PrintableString encoding of a Public Key Thumbprint described in Section 5.4.

When the DnQualifier appears in the Subject name field, it is the thumbprint of the subject public key that appears in this certificate. When the DnQualifier appears in the Issuer name field, it is the thumbprint of the public key that is used to verify the signature on this certificate (i.e., the thumbprint of the public key that appears in the issuer's certificate).

This field is included to solve various security problems that can arise in an architecture that supports multiple root certificates.

5.3.2 Root Name (OrganizationName)

The specification in this document implies that there will be multiple roots of trust for naming entities. The OrganizationName identifies the entity that is responsible for the root of trust for this certificate.

Exactly one instance of the OrganizationName attribute is required in the Subject name and the Issuer name. It shall be a PrintableString. It should be a meaningful (to humans) name of the organization that is providing the root of trust for all certificates in this chain. There may be multiple roots of trust. The OrganizationName in the Issuer field shall match the OrganizationName in the Subject field. This means that the OrganizationName shall be the same in all certificates that chain back to the same root.

The OrganizationName attribute shall be unique. Vendors can choose their own value for this field as long as it does not match that of another vendor. The values of this field should be chosen to be sufficiently distinct that a human would not confuse two similar names. This name actually identifies the root of trust for the system that issues certificates, so it is more specific than the name of the organization that owns the issuing system. For example, a name like "DC.CA.BigBlue.Com" would be a better name than "BigBlue.Com". This

name might exist for a very long time, so a company that might be purchased by another company may wish to choose a neutral name to which they have intellectual property rights.

5.3.3 Organization Name (OrganizationUnitName)

There shall be one instance of the OrganizationUnitName attribute in the Subject name and the Issuer name fields. It shall be a PrintableString. Security devices do not perform any checks on this attribute, and OrganizationUnitName is ignored by the certificate validation and chaining rules. It is intended for the benefit of humans and may appear in log records.

The OrganizationUnitName that appears in the Subject name field is the name of the organization to which the certificate has been issued and supplements the vendor information found in the CommonName attribute. The OrganizationUnitName that appears in the Issuer name field is name of the organization that issued the certificate.

5.3.4 Entity Name and Roles (CommonName)

Exactly one instance of this attribute shall appear in the Subject name and the Issuer name fields. It shall be a PrintableString. It expresses the role(s) performed by the entity and expresses the physical identification of the entity (make, model, and serial number for devices).

The CommonName attribute is structured to express the authorized roles of the entity and a description of the entity (see Annex A). The value of this field has two parts separated by the leftmost period (“.”) character. The first part is a list of words (that only contain the 52 upper and lowercase letters) separated by single space characters. Each word indicates a role (see Annex A roles encoding table). The second part is a unique label for the entity.

The Role shall be present in all leaf (end-entity – i.e., exhibition security device) certificates. When the role is omitted, a period character shall be the first character of the CommonName.

5.4 Certificate and Public Key Thumbprint

The Public Key Thumbprint is a statistically unique identifier of a public key, and thus also an identifier of the matching private key.

A Public Key Thumbprint shall be the SHA-1 hash (see [FIPS-180-2]) of the contents of the SubjectPublicKey BIT STRING in the SubjectPublicKeyInfo field (excluding the DER tag, length, and number of unused bits count in the DER header for the BIT STRING). For details on computing this value see [RFC3280] Section 4.2.1.2 option 1. For using the DnQualifier attribute in certificate names, the Public Key Thumbprint may be Base64 encoded (see [Base64]) to create a printable representation.

The Certificate Thumbprint is a computed value that is the SHA-1 hash of the To-Be-Signed portion of the certificate excluding the DER encoding tag and length. The Certificate Thumbprint may be Base64 encoded (see [Base64]) to create a printable representation.

Informative Note: Certificate thumbprints are not subject to the SHA-1 collision risks that require SHA-256 in other Digital Certificate hash operations.

6 Certificate Processing Rules

This section describes the rules for validating certificates and chains of certificates.

Some security devices may choose to not perform chain validation in cases where the device does not have a list of trusted roots for the intended purpose. In these cases, the device may wish to remember the certificate thumbprint as a means of recognizing when it is speaking to the same entity.

Some security devices may not have a clock, and may choose to skip the validity time check on the leaf certificate in the chain.

6.1 Validation Context

Certificates are always validated in a context. The context consists of the following components, any of which may be empty except for the first, which shall be present:

a) A chain containing the certificate being validated

b) A minimum chain length (number of certificates)

c) A desired role

d) An effective time (i.e. time and date)

e) A set of trusted certificates

f) A set of revoked certificate identifiers (issuerName-serialNumber pairs)

g) A set of revoked public key values

The context is used in the validation algorithm as specified in Section 6.2 below. This table summarizes the context-dependent algorithm steps:

	Context element	Algorithm steps
a)	Cert chain	16,17,18,19
b)	Chain length	16
c)	Desired role	8
d)	Effective time	9
e)	Set of trusted root certs	19
f)	Set of revoked certs	12
g)	Set of revoked keys	12

Informative Note: The actual values of the context, and whether each particular context component is required, optional, or prohibited, are dependent on the specific application in which the certificate is being validated. Such application specifications are outside the scope of this document. Refer to informative Annex A regarding application considerations for validation context.

6.2 Validation Rules

To validate a certificate chain, the entity shall perform at least the following steps. These steps do not need to be performed in this order. Additional checks on the behavior of certificate issuing systems are not required for the entity (e.g., ensuring that the serial number is an unsigned integer value that is 64-bits in length or less, or ensuring that the validity dates are properly encoded, or ensuring that the sequence numbers are unique). However, a certificate issuing system might not be trusted unless it performs these checks itself.

1. Parse the certificate with the ASN.1 DER decoding rules and reject the certificate if there are syntax errors or it is not DER encoded. This avoids the need to re-code certificates that were received in BER format in order to verify the signature.

2. If the version field is not X.509v3, reject it.

3. If any unrecognized extensions in the certificate are marked Critical, reject it.

4. If any required fields are missing, reject it.

5. If the CA attribute of the BasicConstraint field is True, check that the PathLenConstraint value is present and either zero or positive. This disallows certificate chains of unbounded length. If the CA attribute of the BasicConstraint field is False, check that the PathLenConstraint field is absent or zero. Reject certificates that violate these rules.

6. Check that the KeyUsage field is present. If the CA attribute of the BasicConstraint field is True, then only the KeyCertSign, and optionally, the cRLsign, flag shall be set, otherwise the keyCertSign and cRLsign cannot be set and either or both of the DigitalSignature and the KeyEncipherment flags shall be set. Reject certificates that violate this rule.

7. If the OrganizationName in the subject and issuer fields do not match, reject it. This is the only name subordination rule that is enforced.

8. If the certificate is a leaf certificate (one where the CA attribute of the BasicConstraint field is False), check that there is at least one role specified in the CommonName. (Note: It is permitted for non-leaf certificates – those with BasicConstraint.CA set to True – to have an empty list of roles, in which case the first character of the CommonName shall be the period character, which marks the end of the role field within the CommonName.) If the validation context includes a desired role, check that this role appears (see Section 6.1 and informative note there-in). D-Cinema security devices should ignore unrecognized roles appearing in the CommonName.

9. If the validation context includes a desired time, check that the desired time is within the validity dates. Informative Note: In most cases the desired time is the current time, but a different time might be used to examine historical or future information. Implementations that do not need to know the current time in order to otherwise comply with their requirements typically will not include a desired time in the validation context and therefore will skip this step.

10. Check that the signature algorithms specified inside and outside of the certificate body match and that both equal sha256withRSAEncryption. Reject certificates that violate this rule.

11. Reject the certificate if the subject's Public Key is not an RSA key with the expected length and exponent.

12. Reject the certificate if the subject's public key is on the list of revoked public keys, or the issuer and serial number of this certificate is on the list of revoked certificates. Note: if revoked keys or certificates are absent from the validation context, the respective test is not performed.

13. Reject the certificate if the computed subject's Public Key Thumbprint after Base64 encoding does not match the value of the DnQualifier attribute in the Subject name field.

14. Lookup the issuer's certificate using the value of the AuthorityKeyIdentifier attribute. If it is not found, reject the certificate.

15. Validate the SignatureValue in the certificate using the issuer's public key. If not valid, reject the certificate.

To validate a chain of certificates, validate each certificate using the steps above, and also perform the following steps on each pairing of the parent (issuer) certificate and the direct child (subject) certificate.

16. Reject the certificate if the certificate chain does not contain at least the number of different certificates specified in the validation context.

    Informative Note: A minimum chain length of three certificates is recommended for equipment identity applications.

17. Reject the certificate if the issuer field in the child certificate does not match the subject name of the parent certificate. This check provides the important security assurance that the hash of the public keys as expressed in the DnQualifier attributes has the expected value.

18. Reject the certificate if the validity dates of the child certificate are not contained within the validity dates of the parent certificate. Specifically, the start date of the child certificate shall be identical to or later than the start date of the parent certificate, and the end date of the child certificate shall be identical to or earlier than the end date of the parent certificate. This step does not require a real-time clock; it is a consistency check between data in the parent and child certificate. Failing this check indicates a problem with a CA.

19. Reject the certificate if the root of this certificate chain does not appear in the list of trusted certificates that have been included in the context for this validation

The remaining paragraphs of this Section 6.2 are Informative

Most of the above certificate processing rules are standard requirements of X.509 implementations. Specifically, the standard rules are: 1, 2, 3, 4, 5, 9, 14, 15, 17, 18, and 19. Rule 6 checks for the KeyUsage values that are acceptable in this design. Rule 7 is a modification of the standard name-subordination rule. Rule 7 allows flexible names to appear in certificates and yet ensures that the OrganizationName from the root certificate appears in all certificates in the chain. This is done because the OrganizationName is being used as a stand-in for more complex certificate policy information.

Rule 8 checks for Role information that is embedded in the subject name field. Rule 10 constrains the signature algorithm to a single acceptable value to avoid the need for implementations to be able to verify different kinds of signatures. Rule 11 enforces that certificates can only carry the appropriate RSA public keys.

Rule 13 adds an important security check for an architecture that has multiple root certificates. In single-root PKI architectures with full name subordination (the name in a child certificate includes all the attributes from the name of its issuer certificate) the issuer name and serial number can be a unique identifier of a certificate. In this architecture, the thumbprint of the public key is added to ensure uniqueness and limit the damage that an attacker can do if he has short-term access to a CA system (i.e., is able to issue a small number of bogus certificates). Basically, the thumbprint strongly binds a public key value to a certificate name. It is not possible to create a certificate with the same name, but a different public key.

Rule 16 encourages device makers to store their root certificate “offline” and only make the private key of an intermediate CA certificate available during manufacturing.

6.3 Human Verification in D-Cinema Applications (Informative)

The following applies only to D-Cinema applications. Security devices should provide mechanisms or procedures that support the following verification steps that can be performed by humans. These mechanisms may be vendor specific.

1. Display the list of trusted root certificates that are active for each purpose supported by each device. The display should include at least the issuer name and the SerialNumber of each certificate.

2. Display the certificate for each device. The display should include at least the issuer name and the SerialNumber of the certificate and the subject name of the certificate. Notice that the subject name includes information about the D-Cinema Roles supported by the device as well as its make, model and serial number, and the thumbprint of its public key.

3. Verify the make, model, and serial number that appear in the certificate. This information should be accessible to visual inspection with only modest effort on the part of a human, and/or appear on a shipping manifest that accompanies the device during installation. This rule is essential for providing the physical identification that is the foundation of the logical identifications performed using digital certificates.

Annex A CommonName Role Descriptions for D-Cinema Applications (Informative)

The following applies only to D-Cinema applications. The CommonName of a D-Cinema certificate consists of two major parts. The first part defines the roles in a D-Cinema security system that the certified device can fulfill. The second part allows users to identify the physical device that should be the certificate's owner.

Role Descriptions

The first part of the D-Cinema certificate CommonName lists the roles of the certified device. The formatting of these roles is required (normative), and is outlined in Section 5.3.4. Examples of role tokens that can be used are shown here and are informative. Definitive requirements for enumerating roles are defined in other documents.[1]

Encoding of Digital Cinema Roles

	Permitted use		
Encoded Role	**Leaf**	**CA**	**Description**
TMS	yes	no	Theater Management System (Screen Management System)
SM	yes	no	Security Manager embedded in a Media Block
MIC	yes	no	Media Block capable of processing a KDM carrying a MIC key
SPB	yes	no	Secure Processing Block providing physical protection
MDI	yes	no	Media Decryptor – Image/Picture
MDA	yes	no	Media Decryptor – Audio/Sound
MDS	yes	no	Media Decryptor - Subtitle
PR	yes	no	Projector
LE	yes	no	Link Encryptor for picture
LD	yes	no	Link Decryptor for picture
FMI	yes	no	Forensic Marker – Image/Picture (watermark or fingerprinting device)
FMA	yes	no	Forensic Marker – Audio/Sound
LS	yes	no	Log Signer
CS	yes	no	Content Signer (or content creator)

CA certificates do not contain any role information.

Device Naming Conventions

Cryptographic security is always based on physical security, so cryptographic identity has to be tied to a physical identity. The purpose of the second part of the CommonName is to identify a physical device in a way that can be inventoried by human visual inspection. For physical devices like Secure Processing Blocks (SPBs) the label should describe the make, model, and unique serial number of the device. A manufacturer may choose to include version-number or revision-level information as a component of the device identifier in the CommonName. When a version upgrade (e.g., a security firmware patch) is installed in such a device, a new certificate (using the same public key) may be included in the upgrade. Security devices may make decisions about the device using this information (e.g., use the make and model number to decide whether to send older format information to the device).

[1] See, e.g., Digital Cinema Initiatives (DCI) Digital Cinema System Specification: http://www.dcimovies.com/

Some proposed D-Cinema trust structures can be facilitated by allowing devices to be ‘classed’ by some of their manufacturing characteristics (manufacturer, model, version, etc.) This allows a ‘trusted class’ to be created that covers “all devices certified by X that have CommonNames with attribute Y”. It is outside the scope of this document to completely describe these trust systems or to endorse them, but a common method of naming devices will facilitate their creation.

Device names begin after the leftmost period character (0x2E) in the CommonName. Device names should be a series of classifications delimited by periods. The most general classification should be first in the sequence. Subsequent classifications should be increasingly specific, until the most specific classification is last in the sequence. (Much like an internet domain name, but in the opposite order.) For example, a Media Decryptor might have the CommonName “MD.AcmeCinema.MD-300.123-456789.v3_2”. This could correspond to a device made by Acme Cinema with a model number of MD-300, a serial number of 123-456789, and a version of 3_2 (3.2, but the period character cannot be reused).

It is important to note that agreements will need to be reached between the users of these trust lists and the manufacturers of devices. A different manufacturer might have a different number of hierarchy levels, or might use different ordering. For example, Acme could choose to place version numbers above serial number in the hierarchy. (Thus facilitating the class creation of “all MD-300s with firmware version 3.2” over the class creation of “MD-300, serial number 123-456789, whatever the version”.)

For certificate issuers, the device name should describe the name and major version number of the system that supports issuing certificates. Security devices may make decisions about the issuer using this information. For example, the AcmeCinema certificate issuer might have a CommonName like “.AcmeCinema.DCIssuer.v1”. Over time, this same name might appear in other certificates if the issuer decides to change its RSA public and private key.

Annex B Design Features and Validation Context Considerations for D-Cinema Applications (Informative)

Design Features

The following applies only to D-Cinema applications. The certificates specified in this document are designed to have the following features.

- There is no need for new business entities (e.g., trusted third parties).
- There can be a very large number of Rights Owners, Distributors, or Exhibitors. They do not need to sign each other's certificates.
- Any vendor can create a DC certificate, but that vendor must use some means that is not represented by the certificates themselves to establish trust with other vendors and organizations. Specifically, there is no single root of trust that authorizes vendors to create DC certificates.
- Each vendor has their own root certificate to issue (through an intermediate CA certificate) the end-entity certificates to the devices made by that vendor.
- DC end-entity certificates identify a unique physical entity and the set of DC roles that it performs.
- Detailed security issues about configuration, software version numbers, access control and privileges are not represented in certificates.
- Certificates are designed to be less than 4096 bytes long to facilitate storage and processing in cryptographic hardware with limited secure memory.

The primary security benefit of having certificate chains that are three or more levels deep is to allow the root certificate and its matching private key to be stored offline and only used rarely to sign new intermediate-level certificates. This greatly reduces the opportunities for the root private key to be compromised. In practice, longer chains do not add much performance overhead because the software that validates them will remember the results of validating intermediate-level certificates, so there is no need to re-validate the whole chain.

Validation Context Considerations

Certificates are always validated in the context of a chain of certificates that lead to a root certificate. The term "leaf certificate" (sometimes called end-entity certificate) means the end of the certificate chain furthest from the root certificate, and it defines the identity of a D-Cinema security device (which is an entity that cannot issue other certificates). All of the non-leaf certificates are called Certificate Authority (CA) certificates because these certificates belong to a certificate issuing authority. Root certificates are self-signed CA certificates.

D-Cinema certificate validation takes place in the context of one or more intended roles. For example, when an Image Media Block Security Manager (SM) connects to a remote Secure Processing Block (SPB) containing a Media Decryptor (MD), it needs to check that the target SPB certificate includes the MD role.

D-Cinema certificate validation also takes place in the context of a desired time. Often the desired time that is checked against the validity dates in a certificate is the current time. In other cases the desired time is the time when a message was created or the time when a certificate was signed.

The context for validating certificates depends on the purpose of the certificate validation. For example, there may be a different trusted root certificate used to authenticate an update to the software of a security device than the list of root certificates that are acceptable for authenticating a Composition Play List.

The primary validation list is:

1. List of root certificates trusted for the given purpose.

Additional validation lists may optionally include:

2. List of certificates revoked for the given purpose.

3. List of public keys revoked for the given purpose.

Below are given several examples of the use of lists. It is anticipated that the D-Cinema industry will have access to multiple databases which will maintain and supply information relevant to supporting “lists”.

One or more lists of revoked certificates may be used for different purposes by Rights Owners or Exhibitors. For example, if a device is transferred from one exhibition facility to another, the first facility may add the certificate to a revocation list to enforce a policy that the device is no longer able to access the resources of the facility. A device’s certificate may also be added to a certificate revocation list if an appropriate authority becomes convinced that a device has been stolen or compromised.

It is possible that device “models” are discovered to have security or operational design flaws. In this case it would be impractical to add all the individual certificates for instances of this model to a certificate revocation list. The certificate standard includes the make and model information in the certificate for each device which can be used to disallow the use of a particular model. In this “model number” approach, fixing the design flaw would need to include issuing a new certificate to each device and including a new model number.

As in lists of revoked certificates, one or more lists of revoked public keys may be used for different purposes. Depending upon the incident, the set of revoked public keys could be driven by the Rights Owner or the Exhibitor. Each entry in the list of revoked public keys should identify the SubjectPublicKeyInfo that has been revoked. Implementations may store the Public Key Thumbprint and use that for comparison testing instead of comparing against the full SubjectPublicKeyInfo.

The certificate for a device may be revoked without revoking the public key. This might happen when a device is transferred to an organization that wants to change some of the information in the certificate such as the OrganizationName or OrganizationUnitName. In this case, the private key has not been compromised and there is no need to revoke the matching public key. It is thus acceptable for the same public key to appear in the new certificate when the private key has not been compromised.

Annex C Example D-Certificate for Use in D-Cinema Applications (Informative)

The following applies only to D-Cinema applications. An example D-Cinema Certificate is illustrated below in the form of an ASN.1 syntax dump of DER encoding. The example contains two columns, separated by ':' (colon) characters. The first column presents an offset into the certificate proper (after the two byte DER preamble), followed by the data value at that offset . The second column shows the ASN.1 syntax element discovered at that offset. Syntax element hierarchical nesting is indicated by enclosing '{' and '}' (curly braces).

```
0000  447: SEQUENCE {
0004  32F:   SEQUENCE {
0008    3:     [0] {
000A    1:       INTEGER 2
        :       }
000D    1:     INTEGER 25
0010    D:     SEQUENCE {
0012    9:       OBJECT IDENTIFIER sha1withRSAEncryption (1 2 840 13549 1 1 5)
        :         (PKCS #1)
001D    0:       NULL
        :       }
001F   77:     SEQUENCE {
0021   15:       SET {
0023   13:         SEQUENCE {
0025    3:           OBJECT IDENTIFIER organizationName (2 5 4 10)
        :             (X.520 id-at (2 5 4))
002A    C:           PrintableString 'DC.Company.Com'
        :           }
        :         }
0038   18:       SET {
003A   16:         SEQUENCE {
003C    3:           OBJECT IDENTIFIER organizationalUnitName (2 5 4 11)
        :             (X.520 id-at (2 5 4))
0041    F:           PrintableString 'CA.DC.Company.Com'
        :           }
        :         }
0052   1D:       SET {
0054   1B:         SEQUENCE {
0056    3:           OBJECT IDENTIFIER commonName (2 5 4 3)
        :             (X.520 id-at (2 5 4))
005B   14:           PrintableString 'Root.CA.DC.Company.Com'
        :           }
        :         }
0071   25:       SET {
0073   23:         SEQUENCE {
0075    3:           OBJECT IDENTIFIER dnQualifier (2 5 4 46)
        :             (X.520 id-at (2 5 4))
007A   1C:           PrintableString '+62TOmhFLtn+1966oRy3Bk9codM='
        :           }
        :         }
        :       }
0098   1E:     SEQUENCE {
009A    D:       UTCTime 15/02/2005 14:09:34 GMT
```

```
00A9    D:       UTCTime 14/02/2010 14:09:34 GMT
         :       }
00B8   73:     SEQUENCE {
00BA   11:       SET {
00BC    F:         SEQUENCE {
00BE    3:           OBJECT IDENTIFIER organizationalUnitName (2 5 4 11)
         :             (X.520 id-at (2 5 4))
00C3    8:           PrintableString 'ACME SM'
         :           }
         :         }
00CD   15:       SET {
00CF   13:         SEQUENCE {
00D1    3:           OBJECT IDENTIFIER organizationName (2 5 4 10)
         :             (X.520 id-at (2 5 4))
00D6    C:           PrintableString 'DC.Company.Com'
         :           }
         :         }
00E4   20:       SET {
00E6   1E:         SEQUENCE {
00E8    3:           OBJECT IDENTIFIER commonName (2 5 4 3)
         :             (X.520 id-at (2 5 4))
00ED   17:           PrintableString 'SM.ACME-SCC1000-500010'
         :           }
         :         }
0106   25:       SET {
0108   23:         SEQUENCE {
010A    3:           OBJECT IDENTIFIER dnQualifier (2 5 4 46)
         :             (X.520 id-at (2 5 4))
010F   1C:           PrintableString 'dBKySBUKehqzk/TWJwmj/KuE3P8='
         :           }
         :         }
         :       }
012D  122:     SEQUENCE {
0131    D:       SEQUENCE {
0133    9:         OBJECT IDENTIFIER rsaEncryption (1 2 840 113549 1 1 1)
         :           (PKCS #1)
013E    0:         NULL
         :         }
0140  10F:       BIT STRING, encapsulates {
0145  10A:         SEQUENCE {
0149  101:           INTEGER
         :             00 E8 13 C1 C0 4E DF C4 A2 05 0D 34 30 73 92 2F
         :             B7 05 33 C6 12 4A 10 9C 73 78 19 07 A5 A8 EC 7D
         :             30 4F 2A 55 70 3B 51 75 81 5F 00 AB FA 57 2E 15
         :             CF 8A 7D E3 27 B0 35 7F E4 CF 41 B3 CA BE 7B BC
         :             1A C4 A3 45 6F E8 AC 24 E2 D6 F5 66 D8 A3 8D 5A
         :             33 5F CA 1D 1C DF F4 93 78 3A 7E D8 8B E3 B4 EF
         :             75 D3 B2 E8 81 9D 22 B8 08 91 AB 0B E4 54 05 35
         :             A7 FE CA C8 37 BD F6 70 F0 EC 57 85 E3 DC 1B CF
         :             4A 45 7D F3 45 FD AC 76 57 63 5D 1C 7D 24 89 1E
         :             94 D0 E3 E5 B0 32 60 94 E1 28 37 54 78 2B 2F 30
         :             D8 8C 0D E9 63 C9 57 70 CA E7 A1 49 55 E9 EB A4
```

```
         :             98 73 5D E3 0B A9 71 B0 0A 4B 14 75 C8 EF E0 9F
         :             14 4C CB 95 72 49 84 E6 7E CC 16 DB 62 46 92 1C
         :             17 52 55 55 9C FC D6 F9 9D 65 8B 1E DF 2E 70 00
         :             E4 CC 74 2C 66 79 F5 D7 71 CE C5 C5 3C 73 B6 26
         :             8B 76 86 65 C5 9D 58 E8 E6 EA 16 6D 71 2D AF AB
         :             6F
024E    3:           INTEGER 65537
         :           }
         :         }
         :       }
0253   E1:     [3] {
0256   DE:       SEQUENCE {
0259   A1:         SEQUENCE {
025C    3:           OBJECT IDENTIFIER authorityKeyIdentifier (2 5 29 35)
         :             (X.509 id-ce (2 5 29))
0261   99:           OCTET STRING, encapsulates {
0264   96:             SEQUENCE {
0267   14:               [0]
         :                 FB AD 93 3A 68 45 2E D9 FE D7 DE BA A1 1C B7
06
         :                 4F 5C A1 D3
027D   7B:               [1] {
027F   79:                 [4] {
0281   77:                   SEQUENCE {
0283   15:                     SET {
0285   13:                       SEQUENCE {
0287    3:                         OBJECT IDENTIFIER
         :                           organizationName (2 5 4 10)
         :                           (X.520 id-at (2 5 4))
028C    C:                         PrintableString 'DC.Company.Com'
         :                         }
         :                       }
029A   18:                     SET {
029C   16:                       SEQUENCE {
029E    3:                         OBJECT IDENTIFIER
         :                           organizationalUnitName (2 5 4 11)
         :                           (X.520 id-at (2 5 4))
02A3    F:                         PrintableString 'CA.DC.Company.Com'
         :                         }
         :                       }
02B4   1D:                     SET {
02B6   1B:                       SEQUENCE {
02B8    3:                         OBJECT IDENTIFIER commonName (2 5 4 3)
         :                           (X.520 id-at (2 5 4))
02BD   14:                         PrintableString
'Root.CA.DC.Company.Com'
         :                         }
         :                       }
02D3   25:                     SET {
02D5   23:                       SEQUENCE {
02D7    3:                         OBJECT IDENTIFIER dnQualifier (2 5 4
46)
```

```
         :                             (X.520 id-at (2 5 4))
02DC   1C:                           PrintableString
'+62TOmhFLtn+1966oRy3Bk9codM='
         :                           }
         :                         }
         :                       }
         :                     }
         :                   }
02FA    1:                    [2] 00
         :                    }
         :                  }
         :                }
02FD   1D:              SEQUENCE {
02FF    3:                OBJECT IDENTIFIER subjectKeyIdentifier (2 5 29 14)
         :                  (X.509 id-ce (2 5 29))
0304   16:                OCTET STRING, encapsulates {
0306   14:                  OCTET STRING
         :                    74 12 B2 48 15 0A 7A 1A B3 93 F4 D6 27 09 A3 FC
         :                    AB 84 DC FF
         :                  }
         :                }
031C    C:              SEQUENCE {
031E    3:                OBJECT IDENTIFIER basicConstraints (2 5 29 19)
         :                  (X.509 id-ce (2 5 29))
0323    1:                BOOLEAN TRUE
0326    2:                OCTET STRING, encapsulates {
0328    0:                  SEQUENCE {}
         :                  }
         :                }
032A    B:              SEQUENCE {
032C    3:                OBJECT IDENTIFIER keyUsage (2 5 29 15)
         :                  (X.509 id-ce (2 5 29))
0331    4:                OCTET STRING, encapsulates {
0333    2:                  BIT STRING 4 unused bits
         :                    '1101'B
         :                  }
         :                }
         :              }
         :            }
         :          }
0337    D:       SEQUENCE {
0339    9:         OBJECT IDENTIFIER sha1withRSAEncryption (1 2 840 113549 1
1 5)
         :           (PKCS #1)
0344    0:         NULL
         :         }
0346  101:       BIT STRING
         :         82 70 EA 05 F4 A5 C2 15 FD 42 B2 37 7A 3B 19 3A
         :         82 3B B0 51 7A DC 3F E3 F0 D6 A8 56 CD 06 CE DF
         :         FC 84 B3 99 A3 D5 62 6D E7 A9 DC E9 46 D9 0E 6C
         :         D5 DB 7B 6D BE E1 E5 D6 29 F4 86 17 2B 11 6C 07
         :         F2 6A 47 DA 74 0D 5D 69 4C 2E 1B D3 74 C7 7D FD
```

```
:      22 BA 68 A7 DB F9 E6 A6 A4 81 3A 98 B1 73 D7 20
:      8B 85 FE F1 92 BC F7 40 26 11 6D 58 23 31 2F 65
:      3B 90 65 AD 87 A5 1F 9E C2 29 C8 31 C0 EE 57 F4
:      32 35 12 E0 DB 8E 8D 91 93 A8 68 5D 8D 07 FC 2E
:      8C F9 43 E4 3E 99 EA 3B AA 2B 62 03 B0 67 86 D9
:      0D 1D A0 E1 41 70 F8 FB 48 D9 14 A7 4C F2 77 A3
:      C2 C0 10 B1 D8 EF EE 23 18 DB 3E B0 1B 81 FB D1
:      F8 DE C0 6D 91 5B 5D 90 7E E3 EC E4 02 BC 28 D6
:      94 06 C5 2D E4 0B B8 C3 25 27 4E 38 1D ED A5 BE
:      D1 60 F3 80 AE 16 2B 44 B7 BF 62 FF 32 B6 96 60
:      64 62 89 9A 62 18 DE 91 A3 53 C2 57 91 3B B4 D5
:    }
```

Bibliography (Informative)

These references are included to provide 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

[CPL] SMPTE ST 429-7:2006, D-Cinema Packaging — Composition Play List

[ETM] SMPTE ST 430-3:2012, D-Cinema Operations — Extra-Theater Message

[Ferguson] “Practical Cryptography” 2003 By Neils Ferguson and Bruce Schneier. Wiley Publishing, Indianapolis Indiana. See:
http://www.amazon.com/exec/obidos/ASIN/0471223573/

[FIPS-140-2] “Security Requirements for Cryptographic Modules” Version 2, May 25, 2001. FIPS-140-2.
http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf

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

[Kaliski] “A Layman’s Guide to a Subset of ASN.1, BER and DER” By Burton Kaliski of RSA Labs. November 1993 http://luca.ntop.org/Teaching/Appunti/asn1.html

[KDM] SMPTE ST 430-1:2017, D-Cinema Operations — Key Delivery Message

[Multi-Prime] “Public key cryptographic apparatus and method” by Collins, Thomas; Hopkins, Dale; Langford, Susan; Sabin, Michael. U.S. Patent #5,848,159

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

[PL] SMPTE Standard ST 429-8, D-Cinema Packaging — Packing List

[Rescorla] “SSL and TLS: Designing and Building Secure Systems”. Eric Rescorla. Addison Wesley Professional. ISBN 0201615983. October 2000.
http://www.amazon.com/exec/obidos/ASIN/0201615983

[SPKI] “SPKI Certificate Theory” by C. Ellison et al. September 1999. See:
http://www.ietf.org/rfc/rfc2693.txt

[ADTF] SMPTE 429-14:2014, D-Cinema Packaging — Aux Data Track File