KDM : AUTHENTICATED PUBLIC

PRÉFACE

Authenticated Public contient les métadonnées utiles et lisibles à tous.

Les informations sont en clair, sans chiffrement, ne nécessite aucune cryptographie, ni aucun certificat privé pour lire les données.

Elle est utilisée pour la gestion commune des KDM :

- soit par les **récepteurs légitimes** : les players en salle de cinéma, les encodeurs-récepteurs dans les laboratoires.
- soit par des **récepteurs tiers** : les TMS, les systèmes de livraisons de DCP et KDM vers les salles; ou encore de simples humains qui veulent et peuvent vérifier quelques informations sans logiciel en particulier.

A L'INTÉRIEUR D'AUTHENTICATEDPUBLIC

Reprenons la partie **AuthenticatedPublic** de notre précédent KDM en ajoutant des aérations pour une meilleure lecture [1] :

```
<AuthenticatedPublic Id="ID_AuthenticatedPublic">

  <MessageId>urn:uuid:324767c2-b6c9-40ab-90a1-cb947849e097</MessageId>
  <MessageType>http://www.smpte-ra.org/430-1/2006/KDM#kdm-key-type</MessageType>
  <AnnotationText language="en">KDM</AnnotationText>
  <IssueDate>2022-10-02T21:07:09+00:00</IssueDate>

  <Signer xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    <ds:X509IssuerName>dnQualifier=\+LLvuYNO4YBJSp9Jjmlv8oippzQ=,CN=.DC.DMS.DC2.SMPTE,OU=DC.DOREMILABS.COM,O=DC2.SMPTE.DOREMILABS.COM</ds:X509IssuerName>
    <ds:X509SerialNumber>643</ds:X509SerialNumber>
  </Signer>

  <RequiredExtensions>
    <KDMRequiredExtensions xmlns="http://www.smpte-ra.org/schemas/430-1/2006/KDM">

      <Recipient>
        <X509IssuerSerial xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
          <ds:X509IssuerName>dnQualifier=BnB0iDJLgyqiWUjn1uqrOy2/DEE=,CN=.US1.DCS.DOLPHIN.DC2.SMPTE,OU=DC.DOREMILABS.COM,O=DC2.SMPTE.DOREMILABS.COM</ds:X509IssuerName>
          <ds:X509SerialNumber>88529450606517722</ds:X509SerialNumber>
        </X509IssuerSerial>
        <X509SubjectName>dnQualifier=1xGSxNEWLq7EwRnJ2EAEBoOTfIY=,CN=LE SPB MD FM SM.IMB-239697.DC.DOLPHIN.DC2.SMPTE,OU=DC.DOREMILABS.COM,O=DC2.SMPTE.DOREMILABS.COM</X5
      </Recipient>

      <CompositionPlaylistId>urn:uuid:1ce461e5-548f-4b91-a70a-60b7bc077fb5</CompositionPlaylistId>
      <ContentTitleText language="en">DCP-INSIDE KDM</ContentTitleText>
      <ContentAuthenticator>86yEpVl81kHxy/8bbkZTeXJcVx4=</ContentAuthenticator>
      <ContentKeysNotValidBefore>2010-01-01T00:00:00+02:00</ContentKeysNotValidBefore>
      <ContentKeysNotValidAfter>2040-12-31T23:59:59+02:00</ContentKeysNotValidAfter>

      <AuthorizedDeviceInfo>
        <DeviceListIdentifier>urn:uuid:8d000cc3-1ac2-4b65-a01e-aeb3f17a0211</DeviceListIdentifier>
        <DeviceListDescription language="en">my device list</DeviceListDescription>
        <DeviceList>
          <CertificateThumbprint>2jmj7l5rSw0yVb/vlWAYkK/YBwk=</CertificateThumbprint>
        </DeviceList>
      </AuthorizedDeviceInfo>

      <KeyIdList>
        <TypedKeyId>
          <KeyType scope="http://www.smpte-ra.org/430-1/2006/KDM#kdm-key-type">MDIK</KeyType>
          <KeyId>urn:uuid:abadcafe-abad-cafe-abad-cafe00000001</KeyId>
        </TypedKeyId>
        <TypedKeyId>
          <KeyType scope="http://www.smpte-ra.org/430-1/2006/KDM#kdm-key-type">MDAK</KeyType>
          <KeyId>urn:uuid:abadcafe-abad-cafe-abad-cafe00000002</KeyId>
        </TypedKeyId>
        <TypedKeyId>
          <KeyType scope="http://www.smpte-ra.org/430-1/2006/KDM#kdm-key-type">MDSK</KeyType>
          <KeyId>urn:uuid:abadcafe-abad-cafe-abad-cafe00000003</KeyId>
        </TypedKeyId>
        <TypedKeyId>
          <KeyType scope="http://www.dolby.com/cp850/2012/KDM#kdm-key-type">MDEK</KeyType>
          <KeyId>urn:uuid:abadcafe-abad-cafe-abad-cafe00000004</KeyId>
        </TypedKeyId>
      </KeyIdList>

      <ForensicMarkFlagList>
          <ForensicMarkFlag>http://www.smpte-ra.org/430-1/2006/KDM#mrkflg-picture-disable</ForensicMarkFlag>
          <ForensicMarkFlag>http://www.smpte-ra.org/430-1/2006/KDM#mrkflg-audio-disable</ForensicMarkFlag>
          <ForensicMarkFlag>http://www.dcimovies.com/430-1/2006/KDM#mrkflg-audio-disable-above-channel-12</ForensicMarkFlag>
      </ForensicMarkFlagList>

    </KDMRequiredExtensions>
  </RequiredExtensions>

  <NonCriticalExtensions/>

</AuthenticatedPublic>
```

Prenons la base de notre **AuthenticatedPublic**:

Tag Name	Description	Obligatoire/Optionnel
MessageId	ID unique et créé à la génération du KDM	**obligatoire**
MessageType	URI SMPTE	**obligatoire**
AnnotationText	Description lisible à propos du contenu du fichier	optionel
IssueDate	Date d'émission du KDM	**obligatoire**
Signer	Bloc permettant d'identifier le créateur du KDM via une empreinte normée de son certificat (voir paragraphe sur les **Digital Certificates** pour comprendre la nomenclature).	**obligatoire**
RequiredExtensions	Coeur des métadonnées du KDM	**obligatoire** [2]
NonCriticalExtensions	Métadonnées pour les extensions propriétaires	optionel

L'élément **RequiredExtensions** [3] intègre un élément **KDMRequiredExtensions** qui est l'ajout apporté par la norme KDM à la norme ETM (**SMPTE 430-3 - ETM**) pour le bloc **AuthenticatedPublic**. C'est lui qui va intégrer tout le coeur de la métadonnées du KDM. Tout ce qui se trouve à l'intérieur de **KDMRequiredExtensions** sera normé dans la norme KDM (**SMPTE 430-1 - KDM**)

L'élément **NonCriticalExtensions** - vide à 99,99% - peut être utilisé pour des extensions propriétaires. À l'heure actuelle, je n'ai jamais vu ce champ complété, mais je laisse une chance de 0,01% au cas où au moins un KDM avec un **NonCriticalExtensions** se balade dans les milliards de KDMs générés ces 15 dernières années.

Ces différents éléments (tags) doivent respecter cet ordre.

UN BREF PASSAGE DU CÔTÉ DE SIGNER

On doit faire un petit passage sur l'élément **Signer** car il est particulier, il donne un **IssuerName** [4] et un **SerialNumber**.

Comme cela, on se dirait que l'empreinte dans **IssuerName** est l'empreinte du certificat public de l'encodeur (son **SubjectName**).

Signer est-il normé ?

Signer est une instance XML de **KeyInfoType** et respecte la norme xmldsig-core, notamment le passage KeyInfo et son schéma xmldsig-core-schema.xsd

Les différents tags sont utilisés dans les différents XML suivants :

KeyInfoType → X509Data :

- X509IssuerSerial :
  - KDM→Recipient, KDM→Signature→KeyInfo
  - PKL→Signer, PKL→Signature→KeyInfo
  - CPL→Signer, CPL→Signature→KeyInfo
- X509SubjectName :
  - KDM→Recipient
  - PKL→Signer, PKL→Signature→KeyInfo
  - CPL→Signer
- X509Certificate :
  - KDM→Signature→KeyInfo
  - PKL→Signature→KeyInfo
  - CPL→Signature→KeyInfo

Ce n'est pas le cas, c'est le **IssuerName** du certificat parent, donc l'empreinte de l'**autorité de certification** (Certificate Authority / CA).

Voyons les métadonnées du certificat public de l'encodeur :

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 643 (0x283)
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: O = DC2.SMPTE.DOREMILABS.COM, OU = DC.DOREMILABS.COM, CN = .DC.DMS.DC2.SMPTE, dnQualifier = "+LLvuYNO4YBJSp9Jjmlv8oippzQ="
        Validity
            Not Before: Jan  1 00:00:00 2007 GMT
            Not After : Dec 31 23:59:59 2025 GMT
        Subject: O = DC2.SMPTE.DOREMILABS.COM, OU = DC.DOREMILABS.COM, CN = CS.DMSJP2K-80119.DC.DC2.SMPTE, dnQualifier = SQFYSSqWwjefppqasMJmfdmS6lI=
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:bc:62:a5:fb:33:4f:73:58:2a:6a:03:37:2b:52:
                    (...)
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Basic Constraints: critical
                CA:FALSE
            X509v3 Key Usage:
                Digital Signature, Key Encipherment, Data Encipherment
            X509v3 Subject Key Identifier:
                49:01:58:49:2A:96:C2:37:9F:A6:9A:9A:B0:C2:66:7D:D9:92:EA:52
            X509v3 Authority Key Identifier:
                keyid:F8:B2:EF:B9:83:4E:E1:80:49:4A:9F:49:8E:69:6F:F2:88:A9:A7:34
                DirName:/O=DC2.SMPTE.DOREMILABS.COM/OU=DC.DOREMILABS.COM/CN=.DMS.DC2.SMPTE/dnQualifier=RQ\/53RmuLsbzgfPXGlRYmJruwMs=
                serial:02
    Signature Algorithm: sha256WithRSAEncryption
    (...)
```

Pour résumer :

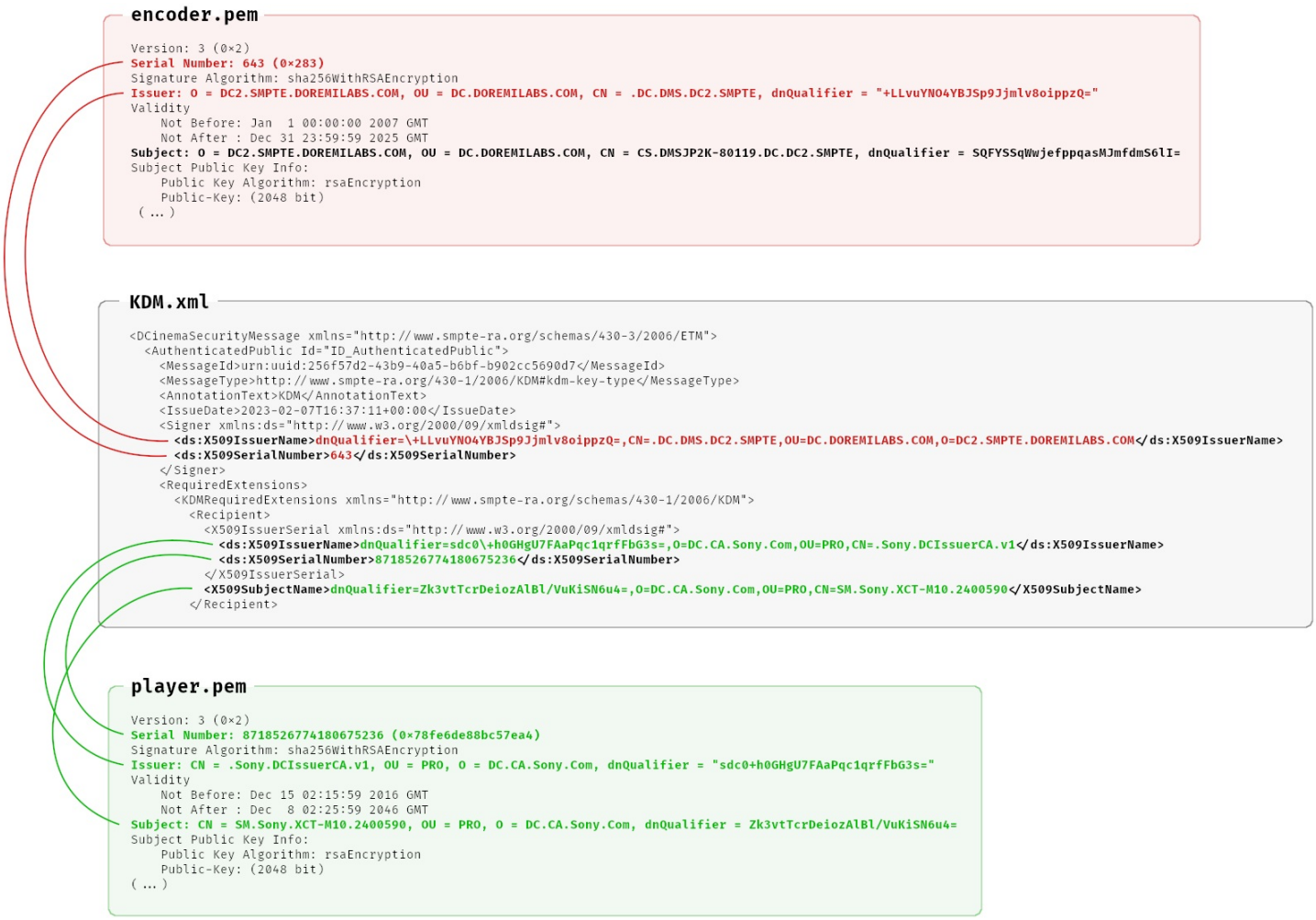
- **Subject** est l'*identificateur* du certificat de l'encodeur.
- **Serial Number** est le numéro de série de ce certificat de l'encodeur.
- **Issuer** est l'*identificateur* du certificat parent qui a signé ce certificat.

Si nous regardons le **Signer** d'un KDM signé avec cet encodeur :

```
<Signer xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    <ds:X509IssuerName>dnQualifier=\+LLvuYNO4YBJSp9Jjmlv8oippzQ=,CN=.DC.DMS.DC2.SMPTE,OU=DC.DOREMILABS.COM,O=DC2.SMPTE.DOREMILABS.COM</ds:X509IssuerName>
    <ds:X509SerialNumber>643</ds:X509SerialNumber>
</Signer>
```

On trouve **IssuerName** et **SerialNumber** mais jamais **SubjectName** qui est la véritable identité du certificat public de l'encodeur.

Voici une image qui permet de mieux voir les interactions entre un **KDM** - et le **certificat de l'encodeur** et le **certificat du player** :



Restrictions des certificats Feuille (Leaf)

Les certificats peuvent avoir des usages et des contraintes.
Si vous voulez en savoir plus, reportez-vous au chapitre Certificats.

À noter que tous les certificats se retrouveront dans la partie **Signature** > **KeyInfo**.

A L'INTÉRIEUR DE KDMREQUIREDEXTENSIONS

L'élément **KDMRequiredExtensions** [5] intègre la grande majorité des métadonnées du KDM : c'est la partie la plus importante de ce bloc.

L'élément **KDMRequiredExtensions** est un élément ajouté par la norme KDM à la norme ETM (**SMPTE 430-3 - ETM**) pour le bloc **AuthenticatedPublic**. C'est lui qui va intégrer tout le coeur de la métadonnées du KDM. Tout ce qui se trouve à l'intérieur de **KDMRequiredExtensions** sera normé dans la norme KDM (**SMPTE 430-1 - KDM**)

Cette partie incorpore :

- Les informations sur le destinataire du KDM (**Recipient**)
- L'identifiant de la CPL qui servira pour le KDM (**CompositionPlaylistId**)
- Le nom de la CPL en question (**ContentTitleText**)
- Les dates de validités du KDM (**ContentKeysNotValidBefore**, **ContentKeysNotValidAfter**)
- La liste des identifiants des clefs et des types des assets associés (**KeyIdList**)
- Des options pour le KDM (**ForensicMarkFlagList**)

Voyons chaque partie une à une.

RECIPIENT: LES INFORMATIONS SUR LE DESTINATAIRE

Recipient identifie **principalement** le récepteur du KDM via des informations provenant des certificats RSA X.509 :

- C'est ce récepteur qui possède le certificat privé qui permettra de déchiffrer les parties chiffrées du KDM (voir chapitre **AuthenticatedPrivate**).
- C'est donc la clef publique de ce récepteur qui va servir pour le chiffrement de la partie **AuthenticatedPrivate**.

Vous allez me dire pourquoi **principalement**, car le bloc **Recipient** intègre l'identité du certificat de l'encodeur mais également l'identité du certificat parent, celui qui a généré le certificat de l'encodeur. C'est donc l'enfant (certificat encodeur) et son parent (autorité de certification) qui seront présent dans le **Recipient**.

Voici un exemple avec de **Recipient** pour un **player DCP Doremi** :

```
<Recipient>
    <X509IssuerSerial xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
        <ds:X509IssuerName>dnQualifier=BnB0iDJLgyqiWUjn1uqrOy2/DEE=,CN=.US1.DCS.DOLPHIN.DC2.SMPTE,OU=DC.DOREMILABS.COM,O=DC2.SMPTE.DOREMILABS.COM</ds:X509IssuerName>
        <ds:X509SerialNumber>88529450606517722</ds:X509SerialNumber>
    </X509IssuerSerial>
    <X509SubjectName>dnQualifier=1xGSxNEWLq7EwRnJ2EAEBoOTfIY=,CN=LE SPB MD FM SM.IMB-239697.DC.DOLPHIN.DC2.SMPTE,OU=DC.DOREMILABS.COM,O=DC2.SMPTE.DOREMILABS.COM</X509Su
</Recipient>
```

Voici un autre exemple pour un **player DCP Sony** :

```
<Recipient>
    <X509IssuerSerial xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
        <ds:X509IssuerName>dnQualifier=sdc0\+h0GHgU7FAaPqc1qrfFbG3s=,O=DC.CA.Sony.Com,OU=PRO,CN=.Sony.DCIssuerCA.v1</ds:X509IssuerName>
        <ds:X509SerialNumber>871852677418067523 6</ds:X509SerialNumber>
    </X509IssuerSerial>
    <X509SubjectName>dnQualifier=Zk3vtTcrDeiozAlBl/VuKiSN6u4=,O=DC.CA.Sony.Com,OU=PRO,CN=SM.Sony.XCT-M10.2400590</X509SubjectName>
</Recipient>
```

Voici un autre exemple pour un **player DCP Christie** :

```
<Recipient>
    <X509IssuerSerial xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
        <ds:X509IssuerName>dnQualifier=51kCIGxNmik7js\+N3MPSGt\+ajJ4=,CN=.signer_dcine_christie,OU=Christie Digital Systems,O=ca.christiedigital.com</ds:X509IssuerName>
        <ds:X509SerialNumber>109676</ds:X509SerialNumber>
    </X509IssuerSerial>
    <X509SubjectName>dnQualifier=V4Tuw2ih9WkrBEH/Xlny4ZYNwHI=,CN=SM.Christie.IMB-S2.00000013C935,OU=Christie Digital Systems,O=ca.christiedigital.com</X509SubjectName>
</Recipient>
```

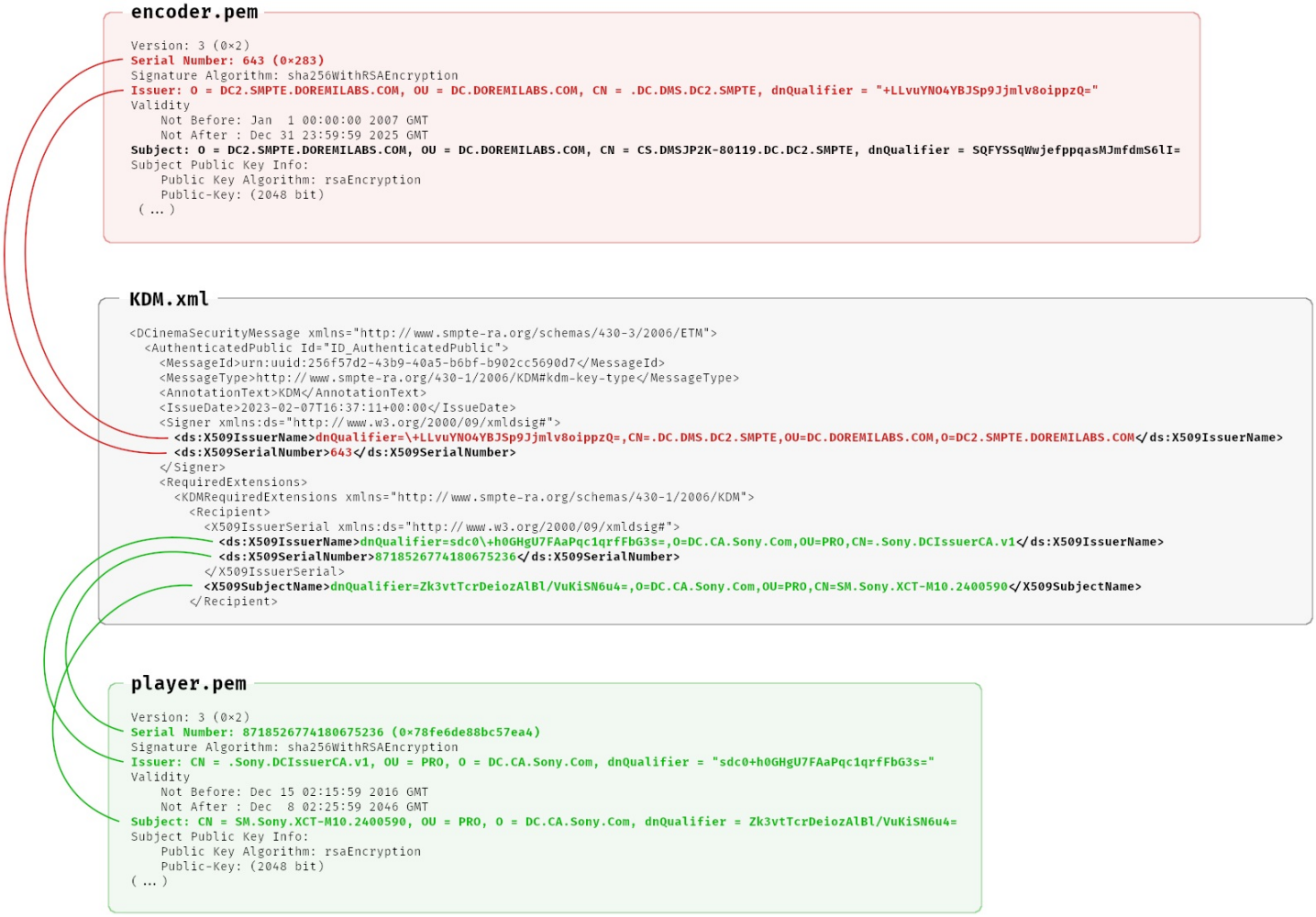
À l'intérieur de **Recipient**, vous aurez deux éléments :

- Le premier élément **X509IssuerSerial** définit l'**autorité de certification** (Certificate Authority ou CA) qui a généré les certificats du récepteur. C'est le parent du certificat, en gros. Nous avons donc son empreinte et son numéro de série.
- Le second élément **X509SubjectName** est l'**empreinte du certificat de notre récepteur**, par exemple de notre player DCP.

Ainsi, le véritable élément qui identifie le player qui doit recevoir ce KDM est le dernier, le **X509SubjectName**.

L'étude sur leurs valeurs sont en dehors du scope de ce chapitre, elles sont étudiées dans le chapitre Certificats, paragraphe KDM et Recipient pour en savoir plus.

Voici une image qui permet de mieux voir les interactions entre un **KDM** - et le **certificat de l'encodeur** et le **certificat du player** :



COMPOSITIONPLAYLISTID : L'IDENTIFIANT DE LA CPL LIÉ AU KDM

Un KDM est toujours lié à une CPL via un identifiant (UUID) inscrit dans le tag `<CompositionPlaylistId>` indiquant l'UUID de la CPL (`<Id>`). Dit autrement :

```
KDM.<CompositionPlaylistId> == CPL.<Id>
```

Il respecte une nomenclature :

- Elle démarre par **urn:uuid**
- Elle respecte la nomenclature d'un **UUID 128-bit**, exemple `1ce461e5-548f-4b91-a70a-60b7bc077fb5`.

Voyons cela avec un exemple de KDM et de CPL côte-à-côte :

KDM

```
<?xml version="1.0" encoding="UTF-8"?>
<DCinemaSecurityMessage (...)>
  <AuthenticatedPublic (...)>
    (...)
    <RequiredExtensions>
      <KDMRequiredExtensions (...)>
        <Recipient>
          <X509IssuerSerial (...)>
            <ds:X509IssuerName>dnQualifier=xxxx=,CN=xxxx,O=xxxxx</ds:X509IssuerName>
            <ds:X509SerialNumber>xxx</ds:X509SerialNumber>
          </X509IssuerSerial>
          <X509SubjectName>dnQualifier=xxxx,CN=xxxx,OU=xxxx,O=xxxx</X509SubjectName>
        </Recipient>
        <CompositionPlaylistId>urn:uuid:1ce461e5-548f-4b91-a70a-60b7bc077fb5</CompositionPlaylistId>
        <ContentTitleText>xxxx</ContentTitleText>
        <ContentKeysNotValidBefore>xxxx</ContentKeysNotValidBefore>
        <ContentKeysNotValidAfter>xxxx</ContentKeysNotValidAfter>
        (...)
```

CPL

```
<?xml version="1.0" encoding="UTF-8"?>
<CompositionPlaylist (...)>
  <Id>urn:uuid:1ce461e5-548f-4b91-a70a-60b7bc077fb5</Id>
  <AnnotationText>xxxx</AnnotationText>
  <IssueDate>xxxx</IssueDate>
  (...)
```

A noter que l'UUID de la CPL se retrouvera également dans chaque **CipherValue** dans le bloc **AuthenticatedPrivate**.

CONTENTTITLETEXT : LE NOM DU KDM

Voici quelques exemples valides :

```
<ContentTitleText>Eiffel_FTR-DVis_S_FR-XX_FR_71-HI-VI-Atmos_4K_PAT_20210630_DGB_SMPTE_OV</ContentTitleText>
<ContentTitleText language="en">Moonlight_FTR_S_EN-XX_51_2K_ALT_20170105_ECL_IOP_OV</ContentTitleText>
<ContentTitleText language="fr">KDM pour le film George</ContentTitleText>
```

Un simple texte lisible par tous et qui sera utilisé par les différents workflows (TMS, Player DCP) pour afficher la teneur du KDM, en général, on Vous êtes libre de

mettre ce que vous voulez, cependant, il est bon de respecter la nomenclature du Digital Cinema Naming Convention.

Notez l'attribut **language** qui permet de définir la langue utilisée grâce à un code respectant la norme ISO-3166.

CONTENTAUTHENTICATOR (OPTIONNEL)

Empreinte de certificat

À COMPLÉTER

CONTENTKEYSNOTVALID : LA GESTION DES DROITS À PEU DE FRAIS

Un KDM a une durée de validité. Un DCP chiffré peut donc être visualisé entre deux dates.

Certains distributeurs (très) paranos font des KDM valides pour seulement quelques heures - ce qui emmerde tout le monde : les laboratoires qui devront faire des KDM à chaque fin de date de validité, et les projectionnistes qui n'ont parfois pas assez de temps pour tester. D'autres sont plus cools et font des dates de validité sur plusieurs mois voire années.

Les deux champs **ContentKeysNotValidBefore** et **ContentKeysNotValidAfter** sont donc les dates de validité du KDM et donc de la plage de disponibilité du DCP:

```
<ContentKeysNotValidBefore>2010-01-01T00:00:00+02:00</ContentKeysNotValidBefore>
<ContentKeysNotValidAfter>2040-12-31T23:59:59+02:00</ContentKeysNotValidAfter>
```

Voici quelques exemples de dates valides :

```
<ContentKeysNotValidAfter>2030-01-01T00:00:00+00:00</ContentKeysNotValidAfter>
<ContentKeysNotValidAfter>2021-10-12T00:00:00+02:00</ContentKeysNotValidAfter>
<ContentKeysNotValidAfter>2040-12-31T23:59:59+00:00</ContentKeysNotValidAfter>
<ContentKeysNotValidAfter>2016-01-02T23:59:00+02:00</ContentKeysNotValidAfter>
<ContentKeysNotValidAfter>2017-01-31T18:55:47+08:00</ContentKeysNotValidAfter>
<ContentKeysNotValidAfter>2017-01-31T18:55:47-05:00</ContentKeysNotValidAfter>
```

Les dates sont normalisées avec la RFC-3339 - Section 5.6 et doivent être de 25 caractères.

Assez rapidement, on constate que :

- Les 3 premiers chiffres indiquent une **année, un mois et un jour**.
- Une séparation "T"
- Les 3 chiffres suivants indiquent **l'heure, la minute et la seconde**.
- Et enfin un symbole + ou -, suivi d'un chiffre qui est l'heure de décalage (timezone) - où la zone géographique UTC.

A noter que ces dates se retrouveront également dans chaque **CipherValue** dans le bloc **AuthenticatedPrivate** (et seront donc chiffrés).

A QUOI SERVENT CES DATES CONCRÈTEMENT ?

La specification DCI impose la possibilité de faire de la gestion de droits (DRM).

En cryptographie, il n'existe aucun algorithme à cryptographie temporelle : cela veut dire qu'il est impossible d'imposer à un cryptogramme un déchiffrement possible seulement à une date ou durant durée.

Ainsi, le déchiffrement sur le **CipherValue** est **illimité dans le temps** : Si vous possédez le certificat privé qui a servi pour le chiffrement des **CipherValue**, vous pouvez déchiffrer les **CipherValue** et récupérer les clefs de déchiffrement **AES** même après la fin de validité officielle du KDM.

Ces dates ne sont respectées que par les devices ayant leurs certifications DCI. Il est important de ne pas oublier cela: ces dates ne sont que des métadonnées qui n'auront aucun impact sur le déchiffrement RSA si vous possédez la clef privée. [6]

AUTHORIZEDDEVICEINFO

À COMPLÉTER

Ce bloc permet de définir plus finement qui a le droit d'utiliser ce KDM. Ce bloc ne joue aucun rôle dans la validation du KDM.

Ce bloc contient 3 éléments :

- **DeviceListIdentifier** : juste un UUID pour l'unicité de la liste.
- **DeviceListDescription** (optionnel) : une simple description lisible par tous.
- **DeviceList** : les empreintes des différents certificats des différents devices qui vont utiliser le KDM. (voir chapitre Certificats)

```
<AuthorizedDeviceInfo>
  <DeviceListIdentifier>urn:uuid:8d000cc3-1ac2-4b65-a01e-aeb3f17a0211</DeviceListIdentifier>
  <DeviceListDescription language="en">my device list</DeviceListDescription>
  <DeviceList>
    <CertificateThumbprint>2jmj7l5rSw0yVb/vlWAYkK/YBwk=</CertificateThumbprint>
  </DeviceList>
</AuthorizedDeviceInfo>
```

KEYIDLIST : LA LISTE DES IDENTIFIANTS DES CLEFS

Ce bloc effectue une simple liste des différents identifiants - ainsi que les types des assets - liés à nos clefs AES.

```
<KeyIdList>
  <TypedKeyId>
    <KeyType scope="http://www.smpte-ra.org/430-1/2006/KDM#kdm-key-type">MDIK</KeyType>
    <KeyId>urn:uuid:abadcafe-abad-cafe-abad-cafe00000001</KeyId>
  </TypedKeyId>
  <TypedKeyId>
    <KeyType scope="http://www.smpte-ra.org/430-1/2006/KDM#kdm-key-type">MDAK</KeyType>
    <KeyId>urn:uuid:abadcafe-abad-cafe-abad-cafe00000002</KeyId>
  </TypedKeyId>
  <TypedKeyId>
    <KeyType scope="http://www.smpte-ra.org/430-1/2006/KDM#kdm-key-type">MDSK</KeyType>
    <KeyId>urn:uuid:abadcafe-abad-cafe-abad-cafe00000003</KeyId>
  </TypedKeyId>
  <TypedKeyId>
    <KeyType scope="http://www.dolby.com/cp850/2012/KDM#kdm-key-type">MDEK</KeyType>
    <KeyId>urn:uuid:abadcafe-abad-cafe-abad-cafe00000004</KeyId>
  </TypedKeyId>
</KeyIdList>
```

Chaque entrée **TypedKeyId** aura son entrée **CipherValue** dans la partie **AuthenticatedPrivate**. Ainsi, si nous avons 4 entrées ici, nous savons que nous aurons 4 **CipherValue** dans **AuthenticatedPrivate**.

Notez qu'ici, ce ne sont pas nos clefs AES, juste des UUIDs présents dans la CPL et simplement répliquées ici pour aider à la gestion par des équipements tiers ne pouvant déchiffrer la partie chiffrée.

Par exemple, si nous ne sommes pas le récepteur (donc nous n'avons pas le certificat privé pour déchiffrer la partie **AuthenticatedPrivate**), nous pouvons contrôler si le KDM possède le même nombre d'entrées de clefs que notre CPL [7] :

```
$ grep "<KeyId>" CPL.xml
  <KeyId>urn:uuid:abadcafe-abad-cafe-abad-cafe00000001</KeyId>
  <KeyId>urn:uuid:abadcafe-abad-cafe-abad-cafe00000002</KeyId>
  <KeyId>urn:uuid:abadcafe-abad-cafe-abad-cafe00000003</KeyId>
  <KeyId>urn:uuid:abadcafe-abad-cafe-abad-cafe00000004</KeyId>

$ grep "<KeyId>" KDM.xml
  <KeyId>urn:uuid:abadcafe-abad-cafe-abad-cafe00000001</KeyId>
  <KeyId>urn:uuid:abadcafe-abad-cafe-abad-cafe00000002</KeyId>
  <KeyId>urn:uuid:abadcafe-abad-cafe-abad-cafe00000003</KeyId>
  <KeyId>urn:uuid:abadcafe-abad-cafe-abad-cafe00000004</KeyId>
```

KEYTYPE : LE TYPE DE L'ASSET LIÉ À LA CLEF

```
<TypedKeyId>
  <KeyType scope="http://www.smpte-ra.org/430-1/2006/KDM#kdm-key-type">MDIK</KeyType>
  <KeyId>urn:uuid:abadcafe-abad-cafe-abad-cafe00000001</KeyId>
</TypedKeyId>
```

Le **KeyType** indique le type de l'asset lié à cette clef et est résumé par un code à 4 lettres.

Le type d'asset est déterminé dans la CPL : par exemple, un **MainPicture** dans une CPL devient un **MDIK** dans un KDM.

Voici les codes définis dans la norme et courants dans les KDM :

Code	Type	Type d'asset dans la CPL
MDIK	Image Essence Key	MainPicture MainSteroscopicPicture
MDAK	Sound Essence Key	MainSound
MDSK	Subtitle Essence Key	MainSubtitle MainCaption MainClosedCaption ClosedCaption ClosedSubtitle
MDEK	AuxData Essence Key	AuxData Enhanced Audio (Immersive Audio / DolbyAtmos / DTS-X / Auro, ...)

Et des codes définis dans la norme mais jamais rencontrés sur le terrain :

Code	Type
FMIK	Image Forensic Marking Key
FMAK	Sound Forensic Marking Key
MDX1	Aux Data Type 1 Key
MDX2	Aux Data Type 2 Key
MDMK	MIC Key

Et les autres assets dans une CPL ?

Sauf si les normes SMPTE évoluent, les autres assets n'ont pas de chiffrement :

- MainMarkers
- CompositionMetadataAsset

Si vous tentez de placer un KeyId dans ces assets au sein de la CPL, vu qu'ils n'ont pas de code car ne contenant pas d'éléments cryptographiques normés, certains encodeurs vont créer un KDM avec ces assets mais leur code resteront vide ou nul (`NULL` , `None` , `Undefined` , ...). D'autres peuvent très bien refuser de créer un KDM.

Ce code est également présent dans la partie **CipherValue**.

KEYID : L'IDENTIFIANT DE LA CLEF DE L'ASSET CHIFFRÉ

```
<TypedKeyId>
  <KeyType scope="http://www.smpte-ra.org/430-1/2006/KDM#kdm-key-type">MDIK</KeyType>
  <KeyId>urn:uuid:abadcafe-abad-cafe-abad-cafe00000001</KeyId>
</TypedKeyId>
```

Les **KeyId** sont les identifiants uniques (UUID) des clefs AES servant à chiffrer les MXF.

Dans un MXF chiffré, cet UUID est inscrit dans le KLV **Cryptographic Context** dans le champ **Cryptographic Key ID**.

Dans chaque CPL, vous avez plusieurs assets, ceux chiffrés comportent un tag **KeyId** contenant un UUID.

Selon la norme KDM, il faut l'intégralité des **KeyId** placés dans une CPL. [6]

Un exemple de partie **AssetList** d'une **CPL** :

```
<!-- Exemple partie CPL -->
  (...)
  <ReelList>
    <Reel>
      <Id>urn:uuid:1db4c5eb-89f5-45e6-87d2-ea8894f9aa5d</Id>
      <AssetList>
        <MainPicture>
          <Id>urn:uuid:3bd3d849-117b-46b0-bc45-3d3228c987c6</Id>
          <EditRate>24 1</EditRate>
          <IntrinsicDuration>24</IntrinsicDuration>
          <EntryPoint>0</EntryPoint>
          <Duration>24</Duration>
          <KeyId>urn:uuid:abadcafe-abad-cafe-abad-cafe00000001</KeyId>
          <Hash>hnBSgENJXOvI6bfpat7GA1VImss=</Hash>
          <FrameRate>24 1</FrameRate>
          <ScreenAspectRatio>4096 2160</ScreenAspectRatio>
        </MainPicture>
        <MainSound>
          <Id>urn:uuid:3433a00f-4bc8-4c16-b33c-b0b0d65711af</Id>
          <EditRate>24 1</EditRate>
          <IntrinsicDuration>24</IntrinsicDuration>
          <EntryPoint>0</EntryPoint>
          <Duration>24</Duration>
          <KeyId>urn:uuid:abadcafe-abad-cafe-abad-cafe00000002</KeyId>
          <Hash>ACd4Aky39E608RNnVfAOisPICZ4=</Hash>
        </MainSound>
      </AssetList>
    </Reel>
  </ReelList>
  (...)
```

Lors d'une création d'un KDM, le générateur de KDM va tout d'abord lire une CPL pour en extraire différents éléments :

- La CPL Id pour en faire la CompositionPlaylistId
- Les différents KeyId de chaque asset,
- (et également le type de l'asset pour le KeyType, vu précédemment)

Il va inscrire chaque **KeyId** dans la **KeyIdList** (partie **AuthenticatedPublic**), puis va piocher dans sa banque de clefs AES pour démarrer le processus de chiffrement RSA - avec des informations en sus - et stocker le tout dans les **CipherValue** (partie **AuthenticatedPrivate**).

Si vous n'avez pas tout compris sur la partie **CipherValue**, pas de panique, on verra cela plus en détail dans le paragraphe **AuthenticatedPrivate**. Il faut juste retenir que les différentes **KeyId** d'une CPL seront intégrés dans notre **KeyIdList** de notre **AuthenticatedPublic**.

FORENSICMARKFLAG : LES OPTIONS DU KDM

```
<!-- Exemple avec beaucoup (trop) de ForensicMarkFlag -->
<ForensicMarkFlagList>
    <ForensicMarkFlag>http://www.smpte-ra.org/430-1/2006/KDM#mrkflg-picture-disable</ForensicMarkFlag>
    <ForensicMarkFlag>http://www.smpte-ra.org/430-1/2006/KDM#mrkflg-audio-disable</ForensicMarkFlag>
    <ForensicMarkFlag>http://www.dcimovies.com/430-1/2006/KDM#mrkflg-audio-disable-above-channel-12</ForensicMarkFlag>
</ForensicMarkFlagList>
```

Les **Forensics Mark Flags** sont des éléments un peu particuliers, ils permettent de contrôler l'activation ou la désactivation de certains paramètres de sécurité (Forensic Mark / Watermarking).

Ils sont utiles dans des cas de figure particuliers liés aux matériels en salle et suivant leurs paramètres, leurs (non-)fonctionnalités ou autres. Certains matériels peuvent ne pas supporter certains types de forensic-mark ou bien refuser de jouer certains assets avec certains forensic-marks.

Par exemple, le **mrkflg-audio-disable-above-channel-12** est utile que pour certains players DCI dans un contexte précis avec un DCP contenant des pistes Dolby Atmos qui pourraient ne pas être correctement jouées (voir le chapitre Immersive Audio, paragraphe KDM, pour plus d'informations).

Les laboratoires doivent être parfaitement au courant du type de **Forensic Mark Flags** à activer suivant le type de récepteur.

Les **Forensic Mark Flags** tendent à être de moins en moins utiliser avec les évolutions de matériel en salle et le support de certaines nouvelles fonctionnalités supportées à chaque mise-à-jour.

Code ForensicMarkFlag	Description
mrkflg-picture-disable	Désactive le forensic-mark pour les assets liés à à la KeyType MDIK (Picture)
mrkflg-audio-disable	Désactive le forensic-mark pour les assets liés à à la KeyType MDAK (utilisé pour DBOX par exemple)
mrkflg-audio-disable-above-channel-<XX>	Désactive le forensic-mark pour les channels de <XX> et supérieurs
mrkflg-audio-disable-above-channel-12	Désactive le forensic-mark du channel 12 à au delà (utile dans certains cas pour l'Immersive Audio / Dolby Atmos)
mrkflg-mdx1-<UUID>-disable	Désactive le forensic-mark pour les KeyType MDX1 et avec comme KeyId <UUID>
mrkflg-mdx2-<UUID>-disable	Désactive le forensic-mark pour les KeyType MDX2 et avec comme KeyId <UUID>
mrkflg-enh-audio-disable	Désactive le forensic-mark pour les KeyType MDEK (utilisé pour le signal de synchronisation de l'Immersive Audio / Dolby Atmos)

ET AVEC (L'ANCIEN) ENCODEUR KDM DE DOREMI ?

Petite interlude historique [8], pour générer des KDM avec (l'ancien) encodeur KDM de Doremi, il faudra fournir un fichier ORCA dont voici quelques paramètres spécifiques pour le Forensic:

```
* Normal
    <ForensicVideo>On</ForensicVideo>
    <ForensicAudio>On</ForensicAudio>

* Atmos
    <FirstForensicAudioDisabledChannel>12</FirstForensicAudioDisabledChannel>
    <ForensicVideo>On</ForensicVideo>
    <ForensicAudio>On</ForensicAudio>

* Dbox
    <FirstForensicAudioDisabledChannel>12</FirstForensicAudioDisabledChannel>
    <ForensicVideo>On</ForensicVideo>
    <ForensicAudio>On</ForensicAudio>

* Disable Audio Watermarking
    <ForensicAudio>Off</ForensicAudio>

    Le KDM aura un flag supplémentaire :
    <ForensicMarkFlag>http://www.smpte-ra.org/430-1/2006/KDM#mrkflg-audio-disable</ForensicMarkFlag>

* Disable Video Watermarking
    <ForensicVideo>On</ForensicVideo>

    Le KDM aura un flag supplémentaire :
    <ForensicMarkFlag>http://www.smpte-ra.org/430-1/2006/KDM#mrkflg-picture-disable</ForensicMarkFlag>
```

NONCRITICALEXTENSIONS : LES EXTENSIONS PROPRIÉTAIRES

C'est la partie utilisée par des extensions propriétaires. Jamais rencontré depuis la création (2005), il me sera difficile d'en dire plus. Si vous avez des KDM possédant des éléments, je suis (très) preneur.

SOUS CHAPITRES

- **KDM - Authenticated Public** : notre partie lisible par tous
- **KDM - Authenticated Private** : notre partie lisible uniquement par le récepteur
- **KDM - Signature** : notre partie pour authentifier et valider le KDM
- **KDM - Codes** : Codes sources et techniques sur les différentes parties du KDM

CHAPITRES CONNEXES

- **La cryptographie** : Pour tout savoir sur la cryptographie utilisée dans le cinéma numérique.
- **Certificats** : Les certificats utilisés dans le cinéma numérique, leurs vies, leurs oeuvres.
- **Cryptographie AES** : La cryptographie symétrique utilisée pour chiffrer les MXF.
- **Cryptographie RSA** : La cryptographie asymétrique utilisée dans les KDM pour chiffrer les clefs AES - entre autres.

NOTES

1. Le format des UUID des KeyId dans les exemples ne respectent pas ««« totalement »»» la norme SMPTE : ↩

   ```
   <KeyId>urn:uuid:abadcafe-abad-cafe-abad-cafe00000001</KeyId>
   ```

   Alors qu'il devrait avoir le **Field Version** à 4:

   ```
   <KeyId>urn:uuid:abadcafe-abad-4afe-abad-cafe00000001</KeyId
   ```

   C'est pour les exemples :)

2. Dans la norme ETM, elle est considérée comme optionnelle, mais avec l'application de la norme KDM, elle devient obligatoire car elle contient des informations essentielles à la bonne gestion du KDM. ↩

3. Dans la norme ETM, elle est considérée comme optionnelle, elle devient obligatoire avec la norme KDM ↩

4. Pourquoi le **IssuerName** à la place de **SubjectName** comme identité du certificat ? Voici quelques pistes normatives : ↩

    - *« 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 »* -- SMPTE 430-3 - KDM
    - *« The issuer name and serial number identify a unique certificate »* -- RFC 2459 - Serial Number
    - Les **KeyUsage** et **BasicConstraint** peuvent empêcher des certificats de pouvoir signer des documents, dont des KDM (SMPTE 430-2 - Certificates)

5. Dans la norme ETM, elle est considérée comme optionnelle, elle devient obligatoire avec la norme KDM ↩

6. C'est un tout petit mensonge pour éviter certaines confusions : les dates font partie des données du **CipherValue**, elles auront donc un impact sur le résultat final lors d'un chiffrement et donc d'un déchiffrement car, dans toute cryptographie un peu sérieuse, chaque bit supplémentaire dans les données à chiffrer entraine un changement irrémédiablement sur l'entièreté du résultat final chiffré. La première affirmation reste vraie dans l'absolue : même après la date de validité, si vous possédez le certificat privé, vous pourrez déchiffrer les différents **CipherValue** sans aucune limitation dans le temps. ↩

7. Notez qu'il est possible de ne pas avoir le même nombre d'entrées, c'est extrêmement rare mais j'en ai déjà rencontré. ↩

    Pas de panique, la norme demande à ce que toutes les clefs soient intégrées dans un seul et même KDM : *« All keys shall be for use with the assets referenced by the Composition Playlist -- SMPTE 430-1 - KDM »*. Vous aurez donc autant d'entrées KeyId des deux côtés.

    Cependant, dans les normes ETM et KDM, il n'existe aucun tag ou élement indiquant qu'un KDM est partiel. Certains encodeurs peuvent mettre en commentaire ce type d'information mais cela reste en dehors des normes et non-interprété par les différents gestionnaires de KDM, players inclus.

    Techniquement, rien n'empêche de générer plusieurs KDMs avec une partie des clefs avec nos propres outils (ou même générer une fausse CPL avec un CPLId déjà existant et que certaines KeyId. Mais là, on tire l'idée à l'extrême, même si techniquement, elle reste valide).

    Enfin, dans les faits, un projectionniste peut recevoir un KDM en amont, avant son DCP et il doit pouvoir quand même ingester ce KDM sans souci. Sans possibilité de comparer avec la CPL liée, il est exclu de refuser un KDM. Après, est-ce que certains players peuvent refuser lorsqu'ils possèdent enfin la CPL ? Je n'ai jamais eu de retour à ce sujet.

8. Depuis le rachat de Doremi par Dolby, l'encodeur KDM Doremi n'est plus vendu. ↩