CERTIFICATS : LA CHAÎNE DE CERTIFICATION

Ce paragraphe est orienté chaîne de certificats x509 DCI.
Nous n'évoquerons donc pas l'étendue des possibilités des chaînes de certificats x509.

Egalement, quand nous utiliserons le terme certificat, nous n'évoquerons que les certificats publics.

PRÉFACE

Comme nous l'avons vu précédemment dans les autres chapitres, sur un appareil DCI (player, encodeur), vous avez un certificat (publique) qui va servir à la gestion cryptographique de certains éléments comme les KDM, les PKL ou les CPL, entre autres.

Malgré son indépendance toute relative, nous avons remarqué que ce certificat était lié à un autre certificat, son "créateur", son direct parent et appelé un **Issuer** : c'est le signataire de ce certificat enfant, c'est lui qui l'a créé et signé.

Ce certificat parent - qui donc a un droit de création de certificat - fait partie d'une **Autorité de Certification** (Certificate Authority ou abbrégé en CA).

Cette hiérarchie est appelée une **chaîne de certification** (ou Certificate Chain).

A QUI APPARTIENT CETTE CHAÎNE DE CERTIFICATION ?

Une chaîne de certification appartient à un "constructeur" : un constructeur peut être un constructeur d'un player, d'un encodeur ou tout type de matériel physique, ou même un développeur de logiciel lié au cinéma numérique, ou encore tout autre type d'entité ayant besoin de gérer une chaîne de certification qui sera utile dans le workflow DCI.

Pour chaque matériel, chaque constructeur va générer des certificats pour chacune de ses machines. Ainsi, par un exemple pour un player DCI, le constructeur va créer un certificat publique pour permettre à des laboratoires de créer des KDM pour cette machine (échange sécurisé) ou pour un encodeur de créer un certificat afin de valider les parties signer d'un KDM, d'une PKL ou d'une CPL avec le certificat d'un encodeur.

A la fin de tout ceci, à la question "à qui appartient cette chaîne de certification", vous pourrez enfin dire "à moi" :)

LA CHAÎNE : DE HAUT EN BAS

Si nous devions résumer, une chaîne de certification respecte une règle en ayant trois parties majeures distinctes :

- Un certificat racine : **Root Certificate**, l'élément le plus haut de la hiérarchie.
- Un ou plusieurs certificats intermédiaires : **Intermediate Certificate**
- Un ou plusieurs certificats en bout de chaîne : **Leaf Certificate**, l'élément le plus bas de la hiérarchie et lié à un matériel.

3 parties seulement ?

La norme SMPTE et le DCI-CTP recommandent très fortement (dans le sens où vous y êtes obligé :) d'avoir

au minimum 3 niveaux de certificats (root, intermediate(s) et leaf(s)) :

- *« a minimum chain length of three certificates is recommended for equipment identity applications. »*
  --- SMPTE 430-2 - Digital Certificate
- *« A complete certificate chain starts with a leaf certificate and ends with a self-signed (CA root) certificate. Between the leaf certificate and the CA root certificate there should be one or more signer certificates. »*
  --- Digital Cinema - Compliant Test Plan

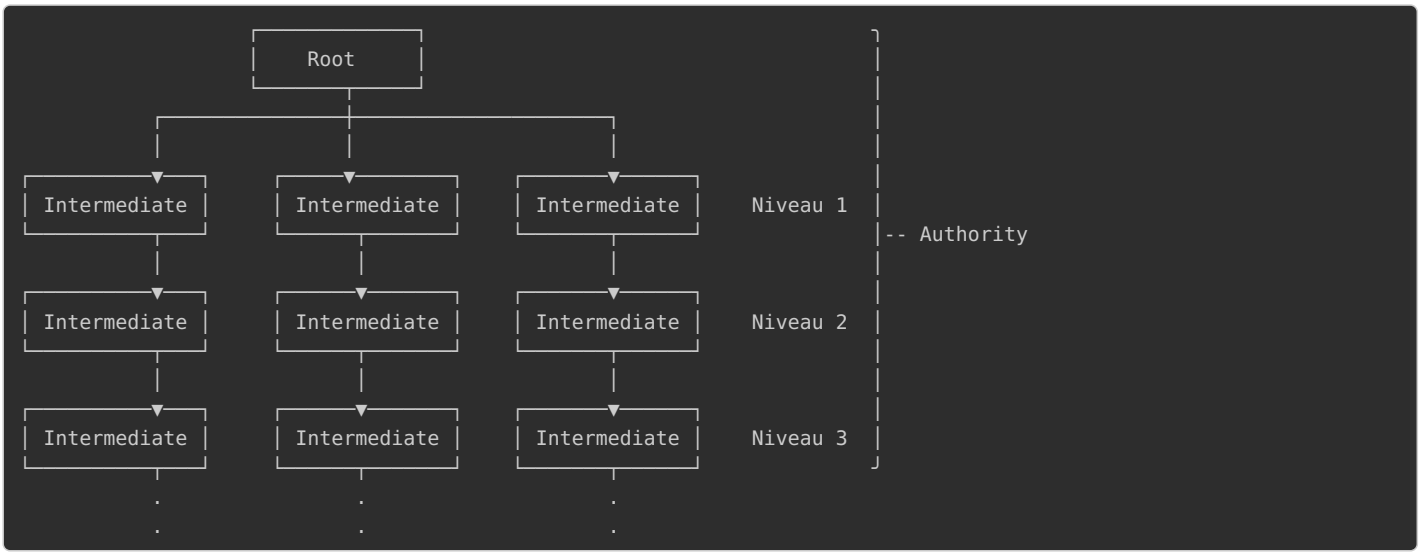
ROOT CERTIFICATE

La chaîne de certification débute par un certificat tout en haut de la hiérarchie, certificat appelé **Root Certificate**. C'est le premier certificat qui a été créé dans la hiérarchie et c'est lui qui va chapoter les certificats en lien direct avec lui-même, juste en dessous.

Ce certificat n'est pas lié à un matériel.

INTERMEDIATE CERTIFICATE

Le certificat parent va permettre la création de sous-branches directes appelées **Certificats Intermédiaires** (ou **Intermediates Certificates**). Ces certificats ont le droit de gérer et créer des certificats directement en dessous d'eux. Les certificats en dessous peuvent être, soit des certificats intermédaires avec les mêmes pouvoirs, soit des certificats avec des droits plus restreints appelés **Leaf Certificate**.



Il n'y a aucune limite (en dehors du PathLen) aux nombres de niveaux dans les certificats intermédiaires.

Ce type de certificat n'est pas lié à un matériel.

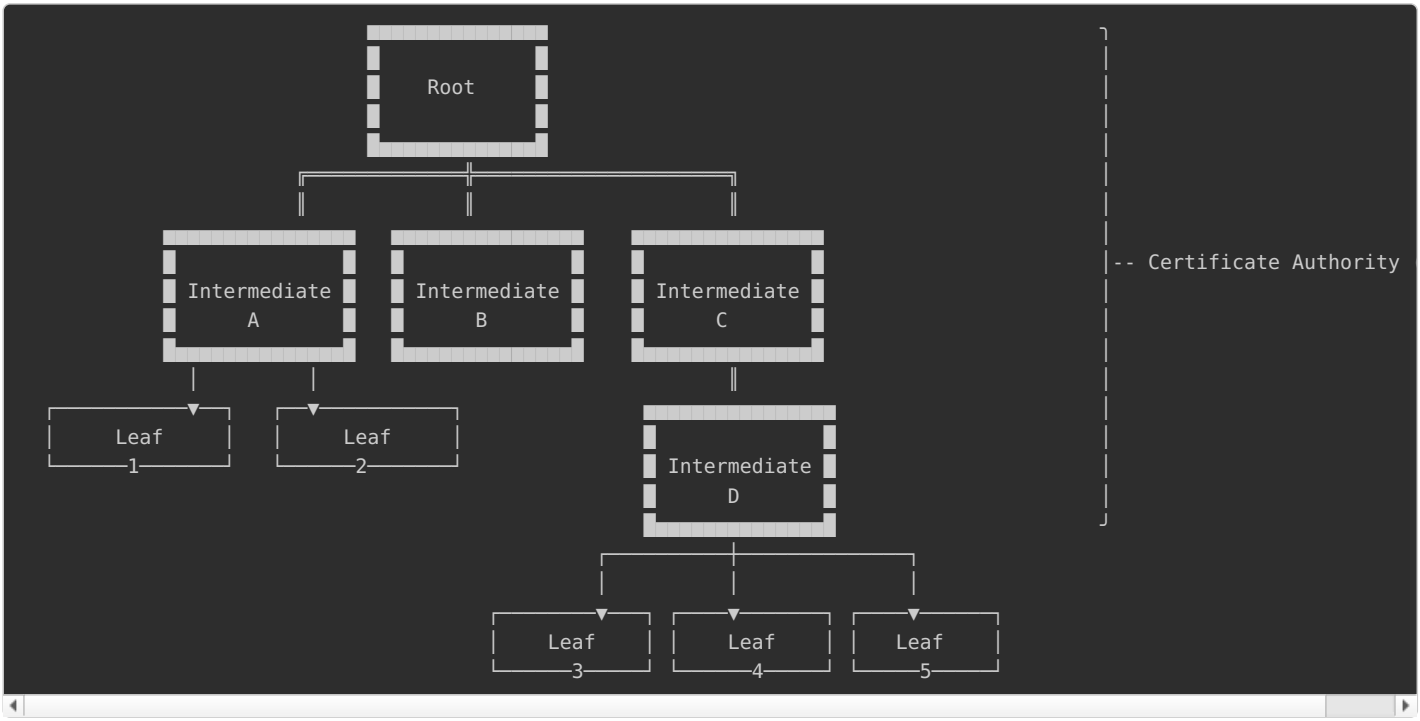
LEAF CERTIFICATE

Enfin, en bout de hiérarchie, ces certificats sont appelés **Leaf Certificates** (feuille en français). Ces certificats ne font pas partie de l'**Autorité de Certification** car ils n'ont pas le droit de créer des certificats enfants. Ces certificats sont utilisés pour et dans les appareils DCI, tels que des players, des encodeurs, etc.

Ce type de certificat est lié à un matériel.

LA HIÉRARCHIE : UN ARBRE INVERSÉ

Voici un exemple parmi d'autres d'une hiérarchie de certificats avec une multitude de certificats **intermédiaires** et de certificats **leafs** :



Ainsi :

- Le certificat **Root** possède 3 certificats enfants (Intermediate A, B et C)
- Le certificat **Intermediate A** possède 2 certificats enfants (Leaf 1 et 2)
- Le certificat **Intermediate B** ne possède aucun certificat enfant
- Le certificat **Intermediate C** possède 1 certificat enfant
- Le certificat **Intermediate D** possède 3 certificats enfants (Leaf 3, 4 et 5)

Les certificats **Root, Intermediate A, B, C, et D** font tous parties de l'**Autorité de Certification**. Tous les certificats Leafs en sont exclus.

LA CHAÎNE SERA STOCKÉE DANS UNE PKL, UNE CPL OU UN KDM

Pour des raisons de logistique, vous retrouverez assez souvent cette chaîne de certificat dans une PKL (signée) ou une CPL (signée) ou dans un KDM (toujours signée).

Cette chaîne de certificat se trouvera toujours dans la partie **Signature** -> **KeyInfo** : le certificat au format PEM dans la partie **X509Certificate** et l'identité de son parent (Issuer) dans la partie **X509IssuerSerial**.

Voici un exemple d'une Signature (certaines parties sont tronquées pour apporter plus de lisibilités) :

```
<ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">

  <ds:SignedInfo>
    (...)
  </ds:SignedInfo>

  <!--
    C'est ici que sont stockés les certificats
    utilisés pour la création du document
    (PKL, CPL ou KDM)
  -->
  <ds:KeyInfo>
      <ds:X509Data>
        <ds:X509IssuerSerial>
          <ds:X509IssuerName>dnQualifier=\+LLvuYNO4YBJSp9Jjmlv8oippzQ=,CN=.DC.DMS.DC2.SMPTE,OU=DC.DOREMILABS.COM,O=D
          <ds:X509SerialNumber>39</ds:X509SerialNumber>
        </ds:X509IssuerSerial>
        <ds:X509Certificate>MIIEfzCCA2egAwIBAgIBJzANBgkqhkiG9w0BAQsFADCBgjEh(...)
(...)qfoeTGQ2Vin9a+udU9/jrSFAvHU2lT8neer722G+CzP/zjFD</ds:X509Certificate>
      </ds:X509Data>
      <ds:X509Data>
        <ds:X509IssuerSerial>
          <ds:X509IssuerName>dnQualifier=RQ/53RmuLsbzgfPXGlRYmJruwMs=,CN=.DMS.DC2.SMPTE,OU=DC.DOREMILABS.COM,O=DC2.SM
          <ds:X509SerialNumber>2</ds:X509SerialNumber>
        </ds:X509IssuerSerial>
        <ds:X509Certificate>MIIEdjCCA16gAwIBAgIBAjANBgkqhkiG9w0BAQsFADB/MSEw(...)
(...)4zsyE0AGuj2nDstRBWLRy+5SlCByTIKp/PL/</ds:X509Certificate>
      </ds:X509Data>
      <ds:X509Data>
        <ds:X509IssuerSerial>
          <ds:X509IssuerName>dnQualifier=pkCB9j5KrAjndhcBkc3fOdfL/BQ=,CN=.PRODUCTS.DC2.SMPTE,OU=DC.DOREMILABS.COM,O=D
          <ds:X509SerialNumber>4</ds:X509SerialNumber>
        </ds:X509IssuerSerial>
        <ds:X509Certificate>MIIEdDCCA1ygAwIBAgIBBDANBgkqhkiG9w0BAQsFADCBhDEh(...)
(...)AAVM+A/J+q+14LJqcElnzHkvNTv2Qvp5+g==</ds:X509Certificate>
      </ds:X509Data>
      <ds:X509Data>
        <ds:X509IssuerSerial>
          <ds:X509IssuerName>dnQualifier=a/wUIHLuFW7RKXpNQGmQ966xOv8=,CN=.ROOT.DC2.SMPTE,OU=DC.DOREMILABS.COM,O=DC2.S
          <ds:X509SerialNumber>2</ds:X509SerialNumber>
        </ds:X509IssuerSerial>
        <ds:X509Certificate>MIIEdjCCA16gAwIBAgIBAjANBgkqhkiG9w0BAQsFADCBgDEh(...)
(...)tIY/GaHgNIVKoTRvgq7YXTVnvGqgolL1zwDx</ds:X509Certificate>
      </ds:X509Data>
      <ds:X509Data>
        <ds:X509IssuerSerial>
          <ds:X509IssuerName>dnQualifier=a/wUIHLuFW7RKXpNQGmQ966xOv8=,CN=.ROOT.DC2.SMPTE,OU=DC.DOREMILABS.COM,O=DC2.S
          <ds:X509SerialNumber>1</ds:X509SerialNumber>
        </ds:X509IssuerSerial>
        <ds:X509Certificate>MIIEcjCCA1qgAwIBAgIBATANBgkqhkiG9w0BAQsFADCBgDEh(...)
(...)Bw/YZzykpHYmuzBxLriBb5TYsWziILI=</ds:X509Certificate>
      </ds:X509Data>
  </ds:KeyInfo>
</ds:Signature>
```

Leurs présences à la fin d'une PKL, d'une CPL ou d'un KDM permettent au destinataire de pouvoir valider de bout en bout la signature des documents, mais attention à l'usurpation de chaîne de certification : rien ne garantit que les différents certificats ajoutés appartiennent au dit-constructeur, il faudra donc vérifier avec la base de certificats du constructeur ou au moins son certificat publique racine.

Usurpation ? un problème de sécurité ?

Mais si on place une chaîne de certification qui n'appartient pas à un constructeur ? que se passe-t-il ?

Effectivement, c'est un point de sécurité que j'avais relevé à la sortie des normes : vu qu'il n'existe aucune centralisation de chaîne de certifications, il est possible pour un attaquant de se faire passer pour une chaîne de certification valide.

Cette absence de centralisation est à la fois positive et négative : positive car tout est décentralisé, les players n'ont pas à se connecter à un serveur central pour vérifier si tel ou tels certificats sont valides. Cela permet également de nouveaux entrants dans le marché des constructeurs et ainsi éviter un monopole.

Également, cela évite de devoir mettre continuellement des mises à jour sur les players si les chaînes de certificats changeaient. Pour le côté négatif, si les laboratoires ne font pas attention, des KDM peuvent être chiffrés avec des certificats n'appartenant pas à un constructeur, non valides, voire pirates.

Un exemple, créons une chaîne de certification qui ressemble à la chaîne de certification Doremi : nous allons créer notre propre certificat Root et ses intermédiaires en reprenant quasiment toutes les structures de chaque certificat Doremi. A la fin, nous allons créer un certificat leaf d'un faux player. Toute la chaîne ressemblera à une chaîne de certification Doremi, sauf qu'elle ne le sera pas (aucune signature ne correspondra, aucun dnQualifier, etc..). Si un opérateur de laboratoire regarde rapidement, il n'y verra que du feu, si le laboratoire ne vérifie pas convenablement ce certificat et sa chaîne de certification (via la base de certificats du constructeur), il peut lancer une procédure de création de KDM et ainsi permettre à un attaquant de récupérer les clefs AES sans aucun souci, vu que l'attaquant est le détenteur de la clef privée rattachée à ce faux certificat public.

Il est donc important de vérifier sur les bases de certificats des différents constructeurs :

Doremi / Dolby	ftp://ftp.cinema.dolby.com	Accès public
Sony	https://spesecure.spe.sony.com	Accès privé
Christie	ftp://certificates.christiedigital.com	Accès privé
XDC	ftp://ftp.xdcinema.com	Accès privé
Qube	ftp://certificates.qubecinema.com	Accès public
GDC	ftp://ftp.gdc-tech.com	Accès privé
Barco	ftp-ssl://certificates.barco.com	Accès privé

Seuls les certificats contenus sur ces sites peuvent être considérés comme valides
(sauf si le site a été piraté et qu'un attaquant a ajouté sa propre chaîne de certification... :)

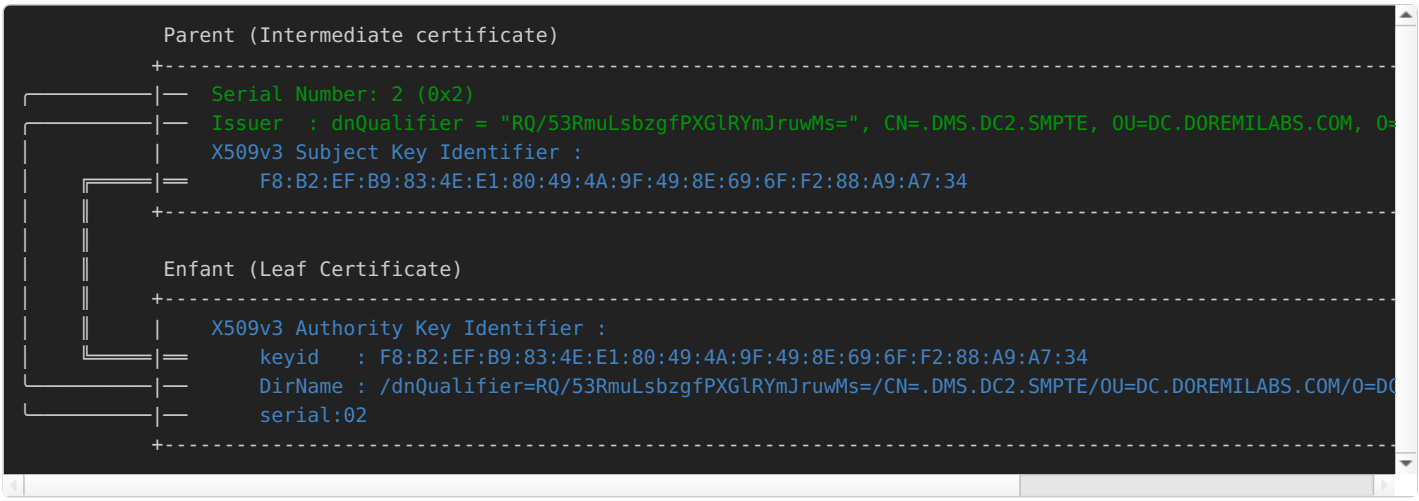
LIENS HIÉRARCHIQUES ENTRE CERTIFICATS

Les certificats se tiennent par des liens hiérarchiques par le biais de certains champs stockés dans le certificat : via certains champs du bloc Metadonnées et via certains champs du bloc x509v3 Extensions :

Déjà, nous avons vu qu'un certificat va adresser son parent via le champ **Issuer**. L'**Issuer** est le **Subject** du certificat parent :

```
     Parent (Intermediate certificate)
    +-------------------------------------------------------------------------------------------------------------------
    |    Issuer  : dnQualifier = "RQ/53RmuLsbzgfPXGlRYmJruwMs=", CN=.DMS.DC2.SMPTE, OU=DC.DOREMILABS.COM, O=DC2.SMPT
┌───|─── Subject : dnQualifier = "+LLvuYNO4YBJSp9Jjmlv8oippzQ=", CN=.DC.DMS.DC2.SMPTE, OU=DC.DOREMILABS.COM, O=DC2.S
│   +-------------------------------------------------------------------------------------------------------------------
│
│    Parent (Intermediate certificate)
│   +-------------------------------------------------------------------------------------------------------------------
└───|─── Issuer  : dnQualifier = "+LLvuYNO4YBJSp9Jjmlv8oippzQ=", CN=.DC.DMS.DC2.SMPTE, OU=DC.DOREMILABS.COM, O=DC2.S
    |    Subject : dnQualifier = "SQFYSSqWwjefppqasMJmfdmS6lI=", CN=CS.DMSJP2K-80119.DC.DC2.SMPTE, OU=DC.DOREMILABS
    +-------------------------------------------------------------------------------------------------------------------
```

Nous avons également un lien hiérarchique dans la partie x509v3 Extensions avec l'**Authority Key Identifier**. Ainsi les différents champs de l'**Authority Key Identifier** du **certificat enfant** font références aux champs **Issuer**, **Subject Key Identifier** et **SerialNumber** du **certificat parent** :



Le champ **Subject Key Identifier** et l'attribut **dnQualifier** sont les empreintes du certificat. Pour plus d'informations, reportez-vous au chapitre Subject Key Identifier ou au chapitre Public Key Thumbprint - dnQualifier

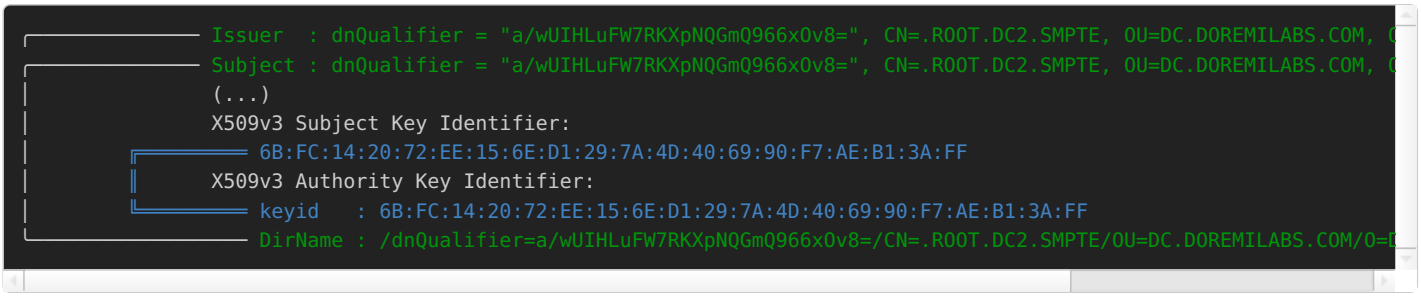
CHAÎNES DE CERTIFICATIONS COMPLET

Chaque certificat fait lien avec son prédécesseur jusqu'au certificat racine.

Un exemple avec une chaîne de certificats Doremi avec 3 intermédiaires, un certificat racine et un certificat leaf :

QUI EST LE PÈRE DU CERTIFICAT RACINE ?

Si vous avez remarqué, le certificat racine s'auto-adresse. Il est lui-même son propre père, son Subject et son Issuer seront les mêmes :



On appelle cela un **certificat self-signed** et la norme oblige à finaliser la chaîne de certificat avec un certificat self-signed : *« A complete certificate chain (..) ends with a self-signed (CA root) certificate. »* et *« A CA Root certificate that is not self-signed shall be cause to fail this test. »* -- CTP

CHAPITRES ANNEXES

- Certificats - Les bases
- Certificats - Les champs (fields) d'un certificat x509 DCI
- Certificats - Identity Attributes - les attributs et leurs rôles
- Certificats - La chaîne de certification ⬅ vous êtes ici
- Certificats - Certificate Thumbprint - l'empreinte du certificat
- Certificats - Public Key Thumbprint (dnQualifier) - l'empreinte de la clef public
- Certificats - Création : Nos propres certificats
- RSA - La cryptographie asymétrique pour le chiffrement et les signatures.