KDM : SIGNATURE

PRÉFACE

La **Signature** intègre des empreintes cryptographiques et des certificats publics.

L'ensemble de la partie **Signature** respecte la norme **XML Digital Signature** (xmldsig) (archive)

A L'INTÉRIEUR DE SIGNATURE

Reprenons la partie **Signature** de notre précédent KDM :

```
<ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
  <ds:SignedInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    <ds:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315#WithComments"/>
    <ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256"/>
    <ds:Reference URI="#ID_AuthenticatedPublic">
      <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
      <ds:DigestValue>qN4dvJpemd94ppazl6ii6nmo9JflBczdpT9yXb3ltow=</ds:DigestValue>
    </ds:Reference>
    <ds:Reference URI="#ID_AuthenticatedPrivate">
      <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
      <ds:DigestValue>9bowz05/W7f4qTJfO4K1VXTYEI14uQgJDYr6Z1uP/Ho=</ds:DigestValue>
    </ds:Reference>
  </ds:SignedInfo>
  <ds:SignatureValue>Fylhfvacb WQ8mcxLiGI3B6HL0EczuISQBYd+Ebkrll4oWs5R
UaKnq4GA6o6+LE2myNf2OdNcJ/7IKPvrDw8NFXR7KVrDWcJa9CGaXd87uxFpsUiBBj3u9Q/E
IM4gaBH/RaaRsy0tKmEenguo6JWMVBBLE20bfLdOrBirpIyTbaIDUCyiUaI4qLrxR0uhuHvJ
gTejDcNbznGPn4esFjcZHTO/C6EDW1U/N3t+AGOcCCjYBf80dIoAOluhVNyglWtVDNeW02sM
tYuWc7m1swzjYqiBk+INkHnPrUvRsxZgzWoo3XGfgbXr15e2TY/IFN2C7bdJ5r6vXpB4dPfH
ThNxmg==</ds:SignatureValue>
  <ds:KeyInfo>
      <ds:X509Data>
        <ds:X509IssuerSerial>
          <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>
        </ds:X509IssuerSerial>
        <ds:X509Certificate>MIIEejCCA2KgAwIBAgICAoMwDQYJKoZIhvcNAQELBQAwgYIx
ITAfBgNVBAoTGERDMi5TTVBURS5ET1JFTUlMQUJTLkNPTTEaMBgGA1UECxMRREMuRE9SRU1JTEFC
Uy5DT00xGjAYBgNVBAMTES5EQy5ETVMuREMyLlNNUFRFMSUwIwYDVQQuExwrTEx2dVlOTzRZQkpT
cDlKam1sdjhvaXBwelE9MB4XDTA3MDEwMTAwMDAwMFoXDTI1MTIzMTIzNTk1OVowgY4xITAfBgNV
BAoTGERDMi5TTVBURS5ET1JFTUlMQUJTLkNPTTEaMBgGA1UECxMRREMuRE9SRU1JTEFCUy5DT00x
JjAkBgNVBAMTHUNTLkRNU0pQMkstODAxMTkuREMuREMyLlNNUFRFMSUwIwYDVQQuExxTUUZZU1Nx
V3dqZWZwcHFhc01KbWZkbVM2bEk9MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAvGKl
+zNPc1gqagM3K1JiThcaQW9u9MOYtTJMTVRixOfu5sGIOYAFf6FmSf5VneEGG9td/iN6GplI1e5u
pTvkyw1CTmiuAp3pqcqLhM8vyPntzloICTNxe9AIiuOnJQO1khLGUQ1pgD5Mvpfya/4Iiqnq8upR
84PoLnnsEKvoxeuXa1iLrCyDZykP7Ybn8U5muzdAqr9YRuFzFzbbq8Fcrz1uPhyKeK0i6+ZQVdfT
8xy4BuNGQcucimPGoImu+2yfNbEcfFQffDA5bGzR2/XDJTLEaqNwe/WXZyGokZtIRzmFJYGb4Zy+
pYGWIK5umuhjNFETufNI48mybNZtelpjIwIDAQABo4HrMIHoMAwGA1UdEwEB/wQCMAAwCwYDVR0P
BAQDAgSwMB0GA1UdDgQWBBRJAVhJKpbCN5+mmpqwwmZ92LlqUjCBqwYDVR0jBIGjMIGggBT4su+5
g07hgElKn0mOaW/yiKmnNKGBhKSBgTB/MSEwHwYDVQQKExhEQzIuU01QVEUuRE9SRU1JTEFCUy5D
T00xGjAYBgNVBAsTEURDLkRPUkVNSUxBQlMuQ09NMRcwFQYDVQQDEw4uRE1TLkRDMi5TTVBURTEl
MCMGA1UELhMcUlEvNTNSbXVMc2J6Z2ZQWEdsUlltSnJ1d01zPTAeFw0wNzAxMDEwMDAwMDBaFw0yNTEyMzEyMzU5NTlaMIGCMSEwHwYDVQQKExhE
AQEAPBpZ6Tkg9TZg2LamLaOCPE6ingBvqmsQHR5V4b99sNU/EvdGL2eJjZFw4YLsxaEmPppIGFdM
IE+QJHrED5YLaJ9i1VvTbD1jNfrclW4Sbc6+mlQKFC6vOD5/go3hLYC8A5w92ea75iX77SuD0DCD
1GIs6FL3EGSrMXC59HFK4ahnIkxcUyhV75Cn14ujaOQpiO+zBxr/Vai/PDZoy6KSm0yYJEh97q4+
vQYQlRWSP1cF9IjLusrYBTjU30f7rygOR4vcqL8xnNQhYprBlJBn+KS3FqNKtrUoHTF0YtLl4I5l
90octl+vtQNGWxq4xU33Du9qXOBXBE9heSfG3CsmVw==</ds:X509Certificate>
      </ds:X509Data>
      <ds:X509Data>
        <ds:X509IssuerSerial>
          <ds:X509IssuerName>dnQualifier=RQ/53RmuLsbzgfPXGlRYmJruwMs=,CN=.DMS.DC2.SMPTE,OU=DC.DOREMILABS.COM,O=DC2.SMPTE.DOREMILABS.COM</ds:X509IssuerName>
          <ds:X509SerialNumber>2</ds:X509SerialNumber>
        </ds:X509IssuerSerial>
        <ds:X509Certificate>MIIEdjCCA16gAwIBAgIBAjANBgkqhkiG9w0BAQsFADB/MSEw
HwYDVQQKExhEQzIuU01QVEUuRE9SRU1JTEFCUy5DT00xGjAYBgNVBAsTEURDLkRPUkVNSUxBQlMu
Q09NMRcwFQYDVQQDEw4uRE1TLkRDMi5TTVBURTElMCMGA1UELhMcUlEvNTNSbXVMc2J6Z2ZQWEds
UlltSnJ1d01zPTAeFw0wNzAxMDEwMDAwMDBaFw0yNTEyMzEyMzU5NTlaMIGCMSEwHwYDVQQKExhE
QzIuU01QVEUuRE9SRU1JTEFCUy5DT00xGjAYBgNVBAsTEURDLkRPUkVNSUxBQlMuQ09NMRowGAYD
VQQDExEuREMuRE1TLkRDMi5TTVBURTElMCMGA1UELhMcK0xMdnVZTk80WUJKU3A5SmptbHY4b2lw
cHpRPTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAMUyBakS7Eb79G3ZFMZ4zOOn/cZ3
qJcck+MAMwjwV8LkK303uBv51cXYNWxalMQQ6h1iPAu6si9RR9XNLGqakHFLUfjt0eC8VJ5/YaC+
DvVq1j9iM5/Jz+CjSqkOlrzXbX92HyMNE0hjCGa4WU6oyRCzFvPZclOI5a2LDKOxC/ngD0LomWjo
MDP7BSkUUzqQnZ2fRuXdOgAfzWm+hBxruWcyEqSGwmV93YBnoHB2WpxVfD+Ztinnp/NJKCadiKf/
sSQpPoqe4j97gY+Zw2DtgT6wddx97ZtCf10QASF5hXgKyCku6O7IlGLroM4TUOldrTClNjR2xSGc
THxcxOltUQcCAwEAAaOB+DCB9TATBgNVHRMBAf8ECTAHAQH/AgIA7TALBgNVHQ8EBAMCAQYwHQYD
VR0OBBYEFPiy77mDTuGASUqfSY5pb/KIqac0MIGxBgNVHSMEgakwgaaAFEUP+d0Zri7G84Hz1xpU
WJia7sDLoYGKpIGHMIGEMSEwHwYDVQQKExhEQzIuU01QVEUuRE9SRU1JTEFCUy5DT00xGjAYBgNV
BAsTEURDLkRPUkVNSUxBQlMuQ09NMRwwGgYDVQQDExMuUFJPRFVDVFMuREMyLlNNUFRFMSUwIwYD
VQQuExxwa0NCOWo1S3JBam5kaGNCa2MzZk9kZkwvQlE9ggEEMA0GCSqGSIb3DQEBCwUAA4IBAQCd
l6G6uFf9+waSrTQqG946Ff1t0b1ANkI6IQQo2sq15aArKjehmrCJzctBfFm3vQj1KlRBI191Mj2m
5Oh/sxcwtG40bSzFeyGZPg9zKA2Cd8oRdw35/8ZLnP96Z/rCxoGNVJAmy9EP8jP9AT/bOk50KvZ/
wo//VN/pscoDrDCaZVLNPAkTdsMNWy+YpV9cnzUGx73jK/iMao++GQ/SEWssBHhjxn3oHWiZi4Xt
ykib8XQqQjyKu4c66jDwu6Ct+MVqUd7ZFPTFListIamu55RnpTXhDKkoN8fIunBsWOs9Kbqbxglx
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=DC2.SMPTE.DOREMILABS.COM</ds:X509IssuerName>
          <ds:X509SerialNumber>4</ds:X509SerialNumber>
        </ds:X509IssuerSerial>
        <ds:X509Certificate>MIIEdDCCA1ygAwIBAgIBBDANBgkqhkiG9w0BAQsFADCBhDEh
MB8GA1UEChMYREMyLlNNUFRFLkRPUkVNSUxBQlMuQ09NMRowGAYDVQQLExFEQy5ET1JFTUlMQUJT
LkNPTTEcMBoGA1UEAxMTLlBST0RVQ1RTLkRDMi5TTVBURTElMCMGA1UELhMccGtDQjlqNUtyQWpu
ZGhjQmtjM2ZPZGZML0JRPTAeFw0wNzAxMDEwMDAwMDBaFw0yNTEyMzEyMzU5NTlaMH8xITAfBgNV
BAoTGERDMi5TTVBURS5ET1JFTUlMQUJTLkNPTTEaMBgGA1UECxMRREMuRE9SRU1JTEFCUy5DT00x
FzAVBgNVBAMTDi5ETVMuREMyLlNNUFRFMSUwIwYDVQQuExxSUS81M1JtdUxzYnpnZlBYR2xSWW1K
cnV3TXM9MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAuqXUXsGFKhmSNFaqAMHloGUp
vTFkFlnjHwRYxUv1JhQjKmysUhyqvXRf7jnx54zsRKdvOXUw3HfGrX9R9gPODL8uVbME406OTqD1
fhhhZ0co02TgXPpaYxkQZlgv9Ygz3pHA51+qXOaMRbPU364gTMPVgXyqljIlDKknZTXeiqp8KB1N
```

```
iTdslrn3IajOxm3pd1Mh2kq2/FyC9qAyZJPbgdcSfm4ZWkxgvllikZ4HQGEuT+P31xmN3K7SQrpk
3jRS1PgFHvUUnCeyQh1jtRlepy2Im4SPpWh1sY+hiR2dakqZCFNKB11wEyZOX3nCqOeDfJL60BNm
9RKgSwPVh6SSywIDAQABo4H0MIHxMBMGA1UdEwEB/wQJMAcBAf8CAgDuMAsGA1UdDwQEAwIBBjAd
BgNVHQ4EFgQURQ/53RmuLsbzgfPXGlRYmJruwMswga0GA1UdIwSBpTCBooAUpkCB9j5KrAjndhcB
kc3fOdfL/BShgYakgYMwgYAxITAfBgNVBAoTGERDMi5TTVBURS5ET1JFTUlMQUJTLkNPTTEaMBgG
A1UECxMRREMuRE9SRU1JTEFCUy5DT00xGDAWBgNVBAMTDy5ST09ULkRDMi5TTVBURTElMCMGA1UE
LhMcYS93VUlITHVGVzdSS1hwTlFHbVE5NjZ4T3Y4PYIBAjANBgkqhkiG9w0BAQsFAAOCAQEAXgI7
+4euCqODivtBpzIunWJBGr+v5jojDLFp6e3fBDznYEY6p4R4B741C7HZ6qE39vV6ZqcBmDjDi7cE
fOa9QmC7Mj0MUM/XXrVlLvHmIRwVcOoZdxz/5JSfQkwPGyXT2MTmgGyTAqeB/NXazjR6YzCOcoqm
CcFRgvubRscN4V5GFC2k7JdvP/s9QeU5cGBePy09sUTm54aFS9MbdQGZa+hqCoNetMvjEGhuUxpe
twOLhXUca9KMT2ikxoA4bNc1P5Ekma8HqX0Dv/BKXtz7YTYUUN2YE12PMaqFLonjvl/mB/n9CHc/
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.SMPTE.DOREMILABS.COM</ds:X509IssuerName>
          <ds:X509SerialNumber>2</ds:X509SerialNumber>
        </ds:X509IssuerSerial>
        <ds:X509Certificate>MIIEdjCCA16gAwIBAgIBAjANBgkqhkiG9w0BAQsFADCBgDEh
MB8GA1UEChMYREMyLlNNUFRFLkRPUkVNSUxBQlMuQ09NMRowGAYDVQQLExFEQy5ET1JFTUlMQUJT
LkNPTTEYMBYGA1UEAxMPLlJPT1QuREMyLlNNUFRFMSUwIwYDVQQuExxhL3dVSUhMdUZXN1JLWHBO
UUdtUTk2NnhPdjg9MB4XDTA3MDEwMTAwMDAwMFoXDTI1MTIzMTIzNTk1OVowgYQxITAfBgNVBAoT
GERDMi5TTVBURS5ET1JFTUlMQUJTLkNPTTEaMBgGA1UECxMRREMuRE9SRU1JTEFCUy5DT00xHDAa
BgNVBAMTEy5QUk9EVUNUUy5EQzIuU01QVEUxJTAjBgNVBC4THHBrQ0I5ajVLckFqbmRoY0JrYzNm
T2RmTC9CUT0wggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDTXmBvZYW1jYsr8GaPzJeD
zeWN+Vus/RDVdaWQ9hLXKw2nLjNFNX+RYB76QiugCGUwBZ71kOL0RQ6kSAkz9FBQGYca9sh2imTc
pEuw/qjbHxClO7DNyd0af5oYtZdopBBy71ksrUWOIFeKgfa1C3iLKJWVeTDZll4s78Xzo0hdQVwF
oFyVRiQRkyr2urj0euMMcykBD5KD+hjKNpxZukzxu8VKKBSJwjifL4SsmV+zsTDJH/nRaySnhLCB
L+aO7MF6d4DF+ANKcD5LX6CQW3NIqPHhdPFC+lAol2/6sPBkbnFEfbWiHbDHS34LJaWaLkqJ3tt/
/ETe6FZe761vkeUbAgMBAAGjgfQwgfEwEwYDVR0TAQH/BAkwBwEB/wICAO8wCwYDVR0PBAQDAgEG
MB0GA1UdDgQWBBSmQIH2PkqsCOd2FwGRzd8518v8FDCBrQYDVR0jBIGlMIGigBRr/BQgcu4VbtEp
ek1AaZD3rrE6/6GBhqSBgzCBgDEhMB8GA1UEChMYREMyLlNNUFRFLkRPUkVNSUxBQlMuQ09NMRow
GAYDVQQLExFEQy5ET1JFTUlMQUJTLkNPTTEYMBYGA1UEAxMPLlJPT1QuREMyLlNNUFRFMSUwIwYD
VQQuExxhL3dVSUhMdUZXN1JLWHBOUUdtUTk2NnhPdjg9ggEBMA0GCSqGSIb3DQEBCwUAA4IBAQCd
sDTq1a3Q96o8ShsyKmEKVUCNnX6ez5XcVg5LXb8QJxXzLD+ffbf15Jj201iElpPQCNU9QHNOOoR9
IoObGZBOLi5ekMhfpfffcpwkwmq3nmb3h6WPLU7PAxLYcE3nNZdbv5QocblNs3e4Ecc3nK2bIS0K
9/e+75UT6wfOCWWCF+AHQt+1Uej6G2f2yZJ5aWqeOv3mqXEtWV8W/iiUEhz6pONdb8Vs0Bhn6FMa
jjTkpSVaC/w7W38htkG0Nk9T5Lf/TdIQHjoALB12suLTZymj8hq53PVxIN+aa0cboVE0PEtwYI5J
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.SMPTE.DOREMILABS.COM</ds:X509IssuerName>
          <ds:X509SerialNumber>1</ds:X509SerialNumber>
        </ds:X509IssuerSerial>
        <ds:X509Certificate>MIIEcjCCA1qgAwIBAgIBATANBgkqhkiG9w0BAQsFADCBgDEh
MB8GA1UEChMYREMyLlNNUFRFLkRPUkVNSUxBQlMuQ09NMRowGAYDVQQLExFEQy5ET1JFTUlMQUJT
LkNPTTEYMBYGA1UEAxMPLlJPT1QuREMyLlNNUFRFMSUwIwYDVQQuExxhL3dVSUhMdUZXN1JLWHBO
UUdtUTk2NnhPdjg9MB4XDTA3MDEwMTAwMDAwMFoXDTI1MTIzMTIzNTk1OVowgYAxITAfBgNVBAoT
GERDMi5TTVBURS5ET1JFTUlMQUJTLkNPTTEaMBgGA1UECxMRREMuRE9SRU1JTEFCUy5DT00xGDAW
BgNVBAMTDy5ST09ULkRDMi5TTVBURTElMCMGA1UELhMcYS93VUlITHVGVzdSS1hwTlFHbVE5NjZ4
T3Y4PTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAJ/xde0jhq2qV0aBlsyv8Cl4vH1+
RRiAl7DyBwKFk0riupyj+/8naVxcWG0Mx9zkWNyDJKowhMqMAMzNzssphxJoFGTOSVDM+WDlS3JZ
BEgdIrvyTjERWMea32nC3uUu8I6JdEhuiUZBHb2FmQBVsp6S3ss9m0iV5zo/zf3Ev2J0j7mDxcp3
Q68zcpmIeik48qpgtfez+svz2tX9eI1mYZW77v9oQiMRZXMGlaiwNlXp3FqkBjAD6MTEWGInEp6b
OtjksVXwZPbFlHlke+wK1wlWMB0wPFAwGsTfeUbTlaaonSHz0JAf0MORYW49cIbInKJffdrZMRBR
tPmsmNDvlIMCAwEAAaOB9DCB8TATBgNVHRMBAf8ECTAHAQH/AgIA8DALBgNVHQ8EBAMCAQYwHQYD
VR0OBBYEFGv8FCBy7hVu0Sl6TUBpkPeusTr/MIGtBgNVHSMEgaUwgaKAFGv8FCBy7hVu0Sl6TUBp
kPeusTr/oYGGpIGDMIGAMSEwHwYDVQQKExhEQzIuU01QVEUuRE9SRU1JTEFCUy5DT00xGjAYBgNV
BAsTEURDLkRPUkVNSUxBQlMuQ09NMRgwFgYDVQQDEw8uUk9PVC5EQzIuU01QVEUxJTAjBgNVBC4T
HGEvd1VJSEx1Rlc3UktYcE5RR21ROTY2eE92OD2CAQEwDQYJKoZIhvcNAQELBQADggEBAH+n8SDy
Hs9JItFnOOgdxKY9/d8iZUJh5GFWdDpInaW6XRoz5dV6vdXPGAiQwvZuKDniWgg4yhrBbesp4RU4
64p2jzL+YUqnUPews7ZT+u2YVbSfh+fO011kSHZLdzTkdA8BWQCVM0e0rCVLRZMw4VPQpwYmnkeM
gheqKfNSyGKiafhuKGqTnqDrr/74/gIq8ZGl/doidoF/DBiiQlYR7eqlMCsTDam+C9E2cpwBN3A1
5yeuD8d+foFVYVmIR71pHFrJ6tRzuRDwqWdVNQa6/kk3pS1Lut7AWpmNirjIhKWJgDW57KlDGIsw
Bw/YZzykpHYmuzBxLriBb5TYsWziILI=</ds:X509Certificate>
      </ds:X509Data>
  </ds:KeyInfo>
</ds:Signature>
```


Cette partie est très touffue, mais si on simplifie pour avoir une vision d'ensemble, voici ce qu'on verrait :

```
<Signature>
    <SignedInfo/>
    <SignatureValue/>
    <KeyInfo/>
</Signature>
```

Cette partie peut être visualisée comme trois parties distinctes :

#	Nom de la partie	Description	Normes
1	**SignedInfo**	intègre les signatures des parties **AuthenticatedPublic** et **AuthenticatedPrivate**	https://www.w3.org/TR/xmldsig-core/#sec-SignedInfo
2	**SignatureValue**	intègre la signature de la partie **SignedInfo**	https://www.w3.org/TR/xmldsig-core/#sec-SignatureValue
3	**KeyInfo**	intègre les **certificats publics RSA utilisés** pour la **SignatureValue**	https://www.w3.org/TR/xmldsig-core/#sec-KeyInfo

Voyons maintenant chaque partie indépendamment en commençant par le premier : **SignedInfo**.

A L'INTÉRIEUR DE SIGNEDINFO

SignedInfo intègre les deux empreintes cryptographiques des parties **AuthenticatedPublic** et **AuthenticatedPrivate** :

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

  <ds:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315#WithComments"/>
  <ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256"/>

  <ds:Reference URI="#ID_AuthenticatedPublic">
    <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
    <ds:DigestValue>qN4dvJpemd94ppazl6ii6nmo9JflBczdpT9yXb3ltow=</ds:DigestValue>
  </ds:Reference>

  <ds:Reference URI="#ID_AuthenticatedPrivate">
    <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
    <ds:DigestValue>9bowz05/W7f4qTJfO4K1VXTYEI14uQgJDYr6Z1uP/Ho=</ds:DigestValue>
  </ds:Reference>

</ds:SignedInfo>
```

En tout début de notre **SignedInfo**, nous avons deux options - **CanonicalizationMethod** et **SignatureMethod** - qui définissent les paramètres pour la création de **SignatureValue**.

CANONICALIZATIONMETHOD : LA MÉTHODE DE CANONISATION DE SIGNEDINFO

L'option **CanonicalizationMethod** indique la méthode de canonisation des données qui va être opérée avant leurs passages cryptographiques.

La valeur dans **CanonicalizationMethod** aura un impact sur **SignatureValue** :

```
<ds:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315#WithComments"/>
```

La canonisation est une méthode pour convertir des jeux de données dans une représentation standardisée ou normalisée.

Dans notre cas, la **CanonicalizationMethod** est directement liée aux données de **SignedInfo** et qui serviront pour créer la **SignatureValue** (que nous verrons dans le paragraphe suivant).

Lors de l'opération de la création de **SignatureValue**, c'est la **première étape** avant le chiffrement :

```
                +-----------------------------------------------------------------------------------------------------------+
                |                                                                                                           |
                |     <ds:SignedInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">                                         |
                |                                                                                                           |
                |         <ds:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315#WithComments"/>     |
                |         <ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256"/>               |
                |                                                                                                           |
                |         <ds:Reference URI="#ID_AuthenticatedPublic">                                                      |
           +----+             <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>                        |
           |    |             <ds:DigestValue>****************************************</ds:DigestValue>                     |
           |    |         </ds:Reference>                                                                                   |
           |    |                                                                                                           |
           |    |         <ds:Reference URI="#ID_AuthenticatedPrivate">                                                     |
           |    |             <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>                        |
           |    |             <ds:DigestValue>****************************************</ds:DigestValue>                     |
           |    |         </ds:Reference>                                                                                   |
           |    |                                                                                                           |
           |    |     </ds:SignedInfo>                                                                                      |
           |    |                                                                                                           |
           |    +-----------------------------------------------------------------------------------------------------------+
           |
           |
           v
++=========================++
||  CanonicalizationMethod  ||
||=========================||
||          C14N           ||
++=========================++
           |
           |
           v
+------------------------+
|     SignatureMethod    |
|------------------------|
|       RSA-SHA256       |
+------------------------+
           |
           v
           |
 +--------------------+
 | Enveloppe / Wrapper |
 |        Base64      |
 +--------------------+
           |
           |        <ds:SignatureValue>********************************************
           |       ************************************************************************
           +--->   ************************************************************************
                   ************************************************************************
                   ************************************************************************
                   ********</ds:SignatureValue>
```

Canonisation = Entonnoir

Voyez cela comme un entonnoir où nous allons pousser tout notre contenu XML qui possède une certaine forme - pas très agréable - pour en sortir un contenu XML avec une forme bien définie et calibrée pour une prochaine étape qui nécessite une certaine forme de données.

Notre canonisation du XML est normé par la norme **C14N** avec l'option **WithComments** et est normalisé par la norme Canonical XML C14N - withComments dont voici les (quelques) principaux critères :

- **Les commentaires sont intégrés dans l'empreinte cryptographie**,
  nous ne pouvons donc ni ajouter, supprimer ou modifier les commentaires dans les parties **SignedInfo**.
- **Aucune tolérance pour les changements des espaces**, des retours à la ligne, ...
  Par exemple, il sera impossible de rajouter un espace entre deux blocs - pour aérer par exemple - sans casser l'empreinte.
- **Tolérance pour les changements des espaces**, des retours à la ligne - dans les **tags** :

Par exemple, `<Id name="test"/>` est équivalent à `<Id name="test" />`

- **Aucune tolérance d'ajouts ou modifications des attributs** dans les tags :
  Par exemple, `<Id name="test"/>` n'est pas équivalent à `<Id name="test" id='test'/>`
- **Les tags réduits seront étendus dans leurs formes longues** :
  Si vous avez un tag `<Foo/>`, lors de la canonisation, il sera transformé en `<Foo></Foo>`.

With Comments ?

L'option **WithComments** dans notre canonisation est importante car - sans cette option - le C14N ignore par défaut totalement les commentaires dans la sortie canonisée. Ici, nous devons prendre en compte les commentaires qui seront de la forme `<!-- comment -->`. Sans cela, notre sortie canonisée sera différente et donc notre empreinte cryptographique erronée.

Pour plus d'informations à propos de la canonisation du XML (C14N), voici quelques références utiles :

Titre	URL
W3C - **Canonicalization XML**	https://www.w3.org/TR/xml-c14n/#WithComments (archive)
RFC 3076 - **Canonicalization XML**	https://www.rfc-editor.org/rfc/rfc3076
Explication de procédure canonisation XML	https://www.di-mgt.com.au/xmldsig-c14n.html
W3C - **Canonicalization XML** en français	http://www.yoyodesign.org/doc/w3c/xml-c14n/
RFC 3076 - **Canonicalization XML** en français	http://abcdrfc.free.fr/rfc-vf/rfc3076.html

SIGNATUREMETHOD : L'ALGORITHME UTILISÉ POUR SIGNATUREVALUE

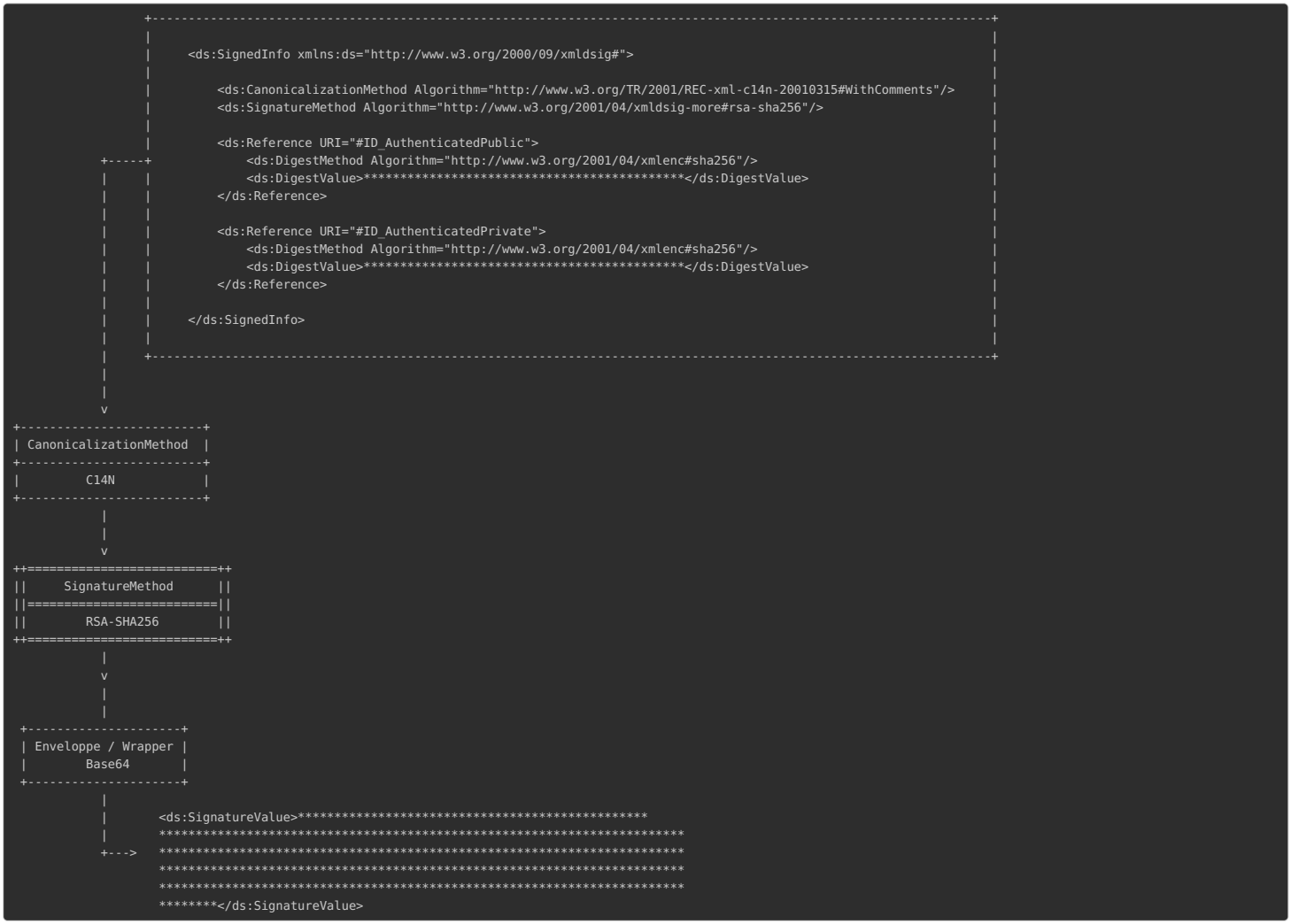
Le **SignatureMethod** indique que la **SignatureValue** (que nous verrons dans le paragraphe suivant) sera formée à l'aide de l'algorithme cryptographique **RSA-SHA256**.

La valeur dans **SignatureMethod** aura un impact sur **SignatureValue** :

```
<ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256"/>
```

Nous utiliserons l'algorithme d'empreinte (Hash) SHA256 et l'algorithme de chiffrement asymétrique RSA - à l'aide d'une clef RSA publique - et enfin, du Base64 en toute fin.

Lors de l'opération de la création de **SignatureValue**, c'est la **seconde étape** après la canonisation des données.



Nous avons vu les deux options nécessaires à la création de **SignatureValue**, voyons maintenant plus en détail l'intérieur de **SignedInfo**, notamment nos **Reference** et leur **DigestValue** respectif.

LES BLOCS REFERENCES ET DIGESTVALUES : LES EMPREINTES DE AUTHENTICATEDPUBLIC ET AUTHENTICATEDPRIVATE

Reprenons nos deux parties XML :

```
<ds:Reference URI="#ID_AuthenticatedPublic">
    <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
    <ds:DigestValue>qN4dvJpemd94ppazl6ii6nmo9JflBczdpT9yXb3ltow=</ds:DigestValue>
</ds:Reference>

<ds:Reference URI="#ID_AuthenticatedPrivate">
    <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
    <ds:DigestValue>9bowz05/W7f4qTJfO4K1VXTYEI14uQgJDYr6Z1uP/Ho=</ds:DigestValue>
</ds:Reference>
```

Focalisons-nous sur la valeur dans nos deux **DigestValue** : Ce sont nos **empreintes cryptographiques** de nos deux blocs **AuthenticatedPublic** et **AuthenticatedPrivate**.

Les attributs **URI** de **Reference** font lien direct soit au bloc **AuthenticatedPublic** soit au bloc **AuthenticatedPrivate** via leurs attributs **Id**, ainsi :

- L'URI `#ID_AuthenticatedPublic` fait référence au bloc possédant l'attribut **Id** `ID_AuthenticatedPublic`.
- L'URI `#ID_AuthenticatedPrivate` fait référence au bloc possédant l'attribut **Id** `ID_AuthenticatedPrivate`.

Un exemple avec **AuthenticatedPublic** :

```
<!-- Partie AuthenticatedPublic -->
<AuthenticatedPublic Id="ID_AuthenticatedPublic">     <----┐
   (...)                                                   |
</AthenticatedPublic>                                      |
                                                           |
<!-- Partie SignedInfo -->                                |
<ds:Reference URI="#ID_AuthenticatedPublic">     <--------┘
    <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
    <ds:DigestValue>qN4dvJpemd94ppazl6ii6nmo9JflBczdpT9yXb3ltow=</ds:DigestValue>
</ds:Reference>
```

Notez bien que l'attribut **URI** n'indique pas le nom du bloc **AuthenticatedPublic** mais l'attribut **Id** possédant la valeur **ID_AuthenticatedPublic**.

Par exemple, la norme XML Signature - Reference nous permettrait de créer un bloc **Foo** avec un Id **Bar** et lier le bloc **Reference** à ce bloc **Foo** (un exemple technique de signature xmlsec hors scope SMPTE est disponible ici) :

Un exemple (non-SMPTE) :

```
<Foo Id="Bar">  <----------------┐
   (...)                         |
</Foo>                           |
                                 |
<ds:Reference URI="#Bar">  <-----┘
   (...)
</ds:Reference>
```

Dans un KDM, les normes nous imposent donc blocs **References** liés à nos **AuthenticatedPublic** et **AuthenticatedPrivate** :

```
<AuthenticatedPublic Id="ID_AuthenticatedPublic">   <----┐
</AuthenticatedPublic>                                  |
                                                        |
<AuthenticatedPrivate Id="ID_AuthenticatedPrivate"> <---|---┐
</AuthenticatedPrivate>                                 |   |
                                                        |   |
                                                        |   |
<ds:Reference URI="#ID_AuthenticatedPublic">  <---------┘   |
</ds:Reference>                                             |
                                                            |
<ds:Reference URI="#ID_AuthenticatedPrivate">  <-------------┘
</ds:Reference>
```

Pour générer un **DigestValue**, nous devons nous référer déjà à option **DigestMethod**. Ici, nous aurons une simple empreinte à l'aide de l'algorithme SHA256.

Comme nous avons vu précédemment, pour pouvoir élaborer un calcul cryptographique sur un contenu XML, nous devons d'abord procéder à une canonisation du XML avant de passer à l'étape de la cryptographie.

Mais dans le cas de **DigestValue**, nous n'avons pas de **CanonicalizationMethod** pour nous indiquer quel type de canonisation nous allons procéder. **CanonicalizationMethod** au dessus (celui de SignedInfo) n'est pas celui qu'on doit utiliser ici.

Dans **Reference**, la méthode de canonisation sélectionnée est implicite : c'est notre **C14N** mais sans les commentaires (**WithoutComments**) : tous les commentaires seront donc exclus de la canonisation.

Ainsi:

```
<Foo>Bar<!-- Boo --></Foo>
```

sera équivalent à

```
<Foo>Bar</Foo>
```

Après avoir canonisé notre XML, nous passons l'ensemble des données dans notre algorithme **SHA256**.

Si nous devions résumer en une ligne de commande la création d'un **DigestValue**, elle serait ainsi :

```
printf "<Foo>Bar</Foo>" \             # notre canonisation XML (manuelle)
    | openssl dgst -sha256 -binary \  # notre empreinte SHA256
    | openssl base64                  # notre enveloppe Base64
```

Nous voila avec notre **DigestValue** de notre **Foo**, il suffit de procéder de la même façon avec **AuthenticatedPublic** ou **AuthenticatedPrivate**.

Vous retrouverez tous les codes et techniques dans la section Codes :

- **Générer un DigestValue** avec OpenSSL
- **XMLSec**: La base des signatures numériques du KDM
- **Générer les Signatures** (DigestValue + SignatureValue) d'un KDM
- **Vérifier les Signatures** (DigestValue + SignatureValue) d'un KDM
- **Canonicalize XML C14N** (Python)

RÉCAPITULATIF DES ENTRÉES ET SORTIES POUR NOTRE DIGESTVALUE :

Les caractéristiques techniques de **DigestValue** [1] :

- **Entrée**
  - Canonicalization XML (C14N without-Comments) **AuthenticatedPublic**
  - Canonicalization XML (C14N without-Comments) **AuthenticatedPrivate**
- **Sortie**
  - **Hash Algorithm** : SHA-256
  - **Encoding Binary Output** : Base64
  - **XML** : https://www.w3.org/TR/xmldsig-core/#sec-DigestValue

RÉCAPITULATIF DES DIFFÉRENTES ÉTAPES POUR DIGESTVALUE :

- XML **AuthenticatedPublic** ou **AuthenticatedPrivate**
- Canonisation XML implicite **C14N-without-Comments**

  Rappel sur cette canonisation XML

  Si nous rajoutons des commentaires dans la partie Authenticated, le hash ne bouge pas.
  Si on rajoute un seul espace ou autres, il bouge. Un tag modifié ou ajouté, même résultat
- **SHA256** (DigestMethod)
- **Base64**

ECRITURE PSEUDO-ALGORITHMIQUE

```
DigestValue = Base64 ( SHA256 ( C14N#withoutComments ( Authenticated XML Block ) )
```

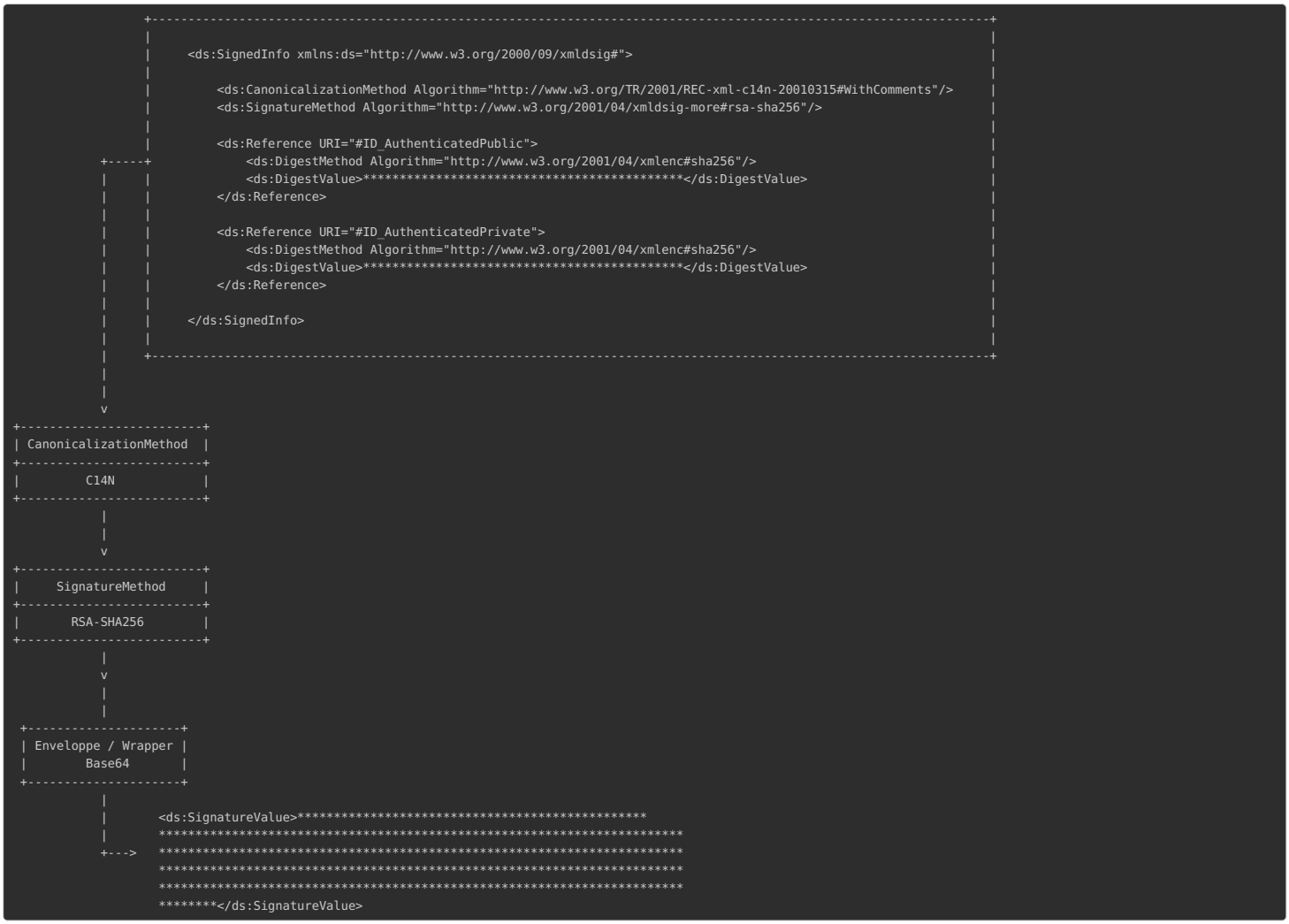
Avec ces deux valeurs de **DigestValue** générées, nous pouvons passer à sa **Signature**.

A L'INTÉRIEUR DE SIGNATUREVALUE

SignatureValue est l'empreinte cryptographique de notre **SignedInfo** incorporant nos deux **DigestValues** complétées.

```
(...)
<ds:SignatureValue>FylhfvacbWQ8mcxLiGI3B6HL0EczuISQBYd+Ebkrll4oWs5RUaKnq4GA6o6+LE2m
yNf2OdNcJ/7IKPvrDw8NFXR7KVrDWcJa9CGaXd87uxFpsUiBBj3u9Q/EIM4gaBH/
RaaRsy0tKmEenguo6JWMVBBLE20bfLdOrBirpIyTbaIDUCyiUaI4qLrxR0uhuHvJ
gTejDcNbznGPn4esFjcZHTO/C6EDW1U/N3t+AGOcCCjYBf80dIoAOluhVNyglWtV
DNeW02sMtYuWc7m1swzjYqiBk+INkHnPrUvRsxZgzWoo3XGfgbXr15e2TY/IFN2C
7bdJ5r6vXpB4dPfHThNxmg==</ds:SignatureValue>
(...)
```

La nature de notre **SignatureValue** sera lié directement aux options dans nos tags **CanonicalizationMethod** et **SignatureMethod** de notre **SignedInfo**, pour rappel :



La **SignatureValue** est le calcul de notre **SignedInfo** - avec ses deux **DigestValues** calculées - canonisé (**C14N**) avec prise en compte des commentaires.

A contrario du **DigestValue** qui va écarter les commentaires XML, la canonisation XML **C14N-WithComments** prend en compte les commentaires XML.

Ainsi :

```
<Foo>Bar<!-- Boo --></Foo>
```

ne sera **pas** l'équivalent de

```
<Foo>Bar</Foo>
```

Sa forme canonisée incorporera donc les commentaires.

Après avoir canonisé **SignedInfo**, le résultat va être hashé avec l'algorithme **SHA256**, puis chiffré (signé) avec l'algorithme **RSA** et la clef privée de l'encodeur. Et enfin être enveloppé à l'aide de l'algorithme **Base64** pour pouvoir être intégré dans son container XML **SignatureValue**.

Si nous devions résumer en une ligne la création d'une **SignatureValue**, elle serait ainsi :

```
printf "<Foo>Bar</Foo>" \
    | openssl dgst \
            -sha256 \
            -sign sample_private_key_non-smpte.pem \
            -binary \
    | openssl base64
eZcLZJMVDScsYc3tJjp+RtRdot0uhe5tT6/nPXkCa6EZuS6nPGpmo6czwESmtmIe
VPRWJNvhD4goToONRjTiT9skuQkwHcIgqBWsoDSvU+KoQRkWz6EN68Y357BEdJgZ
e1tR/LnVblDY6Mk9VjQNHLEOIc2sEE6ZUDDt0geUDD7IJO77bYGdIA5j484rcGEP
QOMHq6aZv0r6DtASgRz0zW4Kl4T3szZrrgiZsEJuutT1mPfBNAoqKkHuO1HygAVg
HYFU7l1mlu5AiSWIFY6t4gF7Ay3JKXYbKHp/jYDqDv7mxnR/PLJeo3cxw086Tpuv
isC9yDwgCWPGZ7jXJfbG9A==
```

Notez que la sortie cryptographique dépendra de notre clef privée RSA.

Si nous devions compléter notre **SignatureValue** avec l'empreinte générée, nous aurions un résultat de la sorte :

```
<ds:SignatureValue>eZcLZJMVDScsYc3tJjp+RtRdot0uhe5tT6/nPXkCa6EZuS6nPGpmo6czwESmtmIe
VPRWJNvhD4goToONRjTiT9skuQkwHcIgqBWsoDSvU+KoQRkWz6EN68Y357BEdJgZ
e1tR/LnVblDY6Mk9VjQNHLEOIc2sEE6ZUDDt0geUDD7IJO77bYGdIA5j484rcGEP
QOMHq6aZv0r6DtASgRz0zW4Kl4T3szZrrgiZsEJuutT1mPfBNAoqKkHuO1HygAVg
HYFU7l1mlu5AiSWIFY6t4gF7Ay3JKXYbKHp/jYDqDv7mxnR/PLJeo3cxw086Tpuv
isC9yDwgCWPGZ7jXJfbG9A==</ds:SignatureValue>
```

Vous retrouverez tous les codes et techniques dans la section Codes :

- **XMLSec**: La base des signatures numériques du KDM
- **Générer les Signatures** (DigestValue + SignatureValue) d'un KDM
- **Vérifier les Signatures** (DigestValue + SignatureValue) d'un KDM
- **Canonicalize XML C14N** (Python)

RÉCAPITULATIF DES ENTRÉES ET SORTIES POUR NOTRE SIGNATUREVALUE :

Caractéristiques techniques de **SignatureValue** [2] :

- **Input** :
  - Canonicalization XML (C14N-WithComments défini par CanonicalizationMethod) de **SignedInfo**
- **Output** :
  - **Hash Algorithm** : SHA-256 [FIPS-180-2]
  - **Padding Algorithm** : PKCS#1 v1.5 (aka RSASSA-PKCS1-v1_5) [3]
  - **Signature Algorithm** : RSA [14]
  - **Encoding Binary Output** : Base64
  - **XML** : https://www.w3.org/TR/xmldsig-core/#sec-SignatureValue

Pour la valeur de **SignatureValue** :

- Pour la **générer** : il faut la **clef privée RSA de l'encodeur** (Signer).
- Pour la **vérifier** : il faut la **clef publique RSA de l'encodeur** (Signer).

RÉCAPITULATIF DES DIFFÉRENTES ÉTAPES POUR SIGNATUREVALUE :

- XML **SignedInfo**
- Canonisation explicite (CanonicalizationMethod) **C14N-With-Comments**

  > **Rappel sur cette canonisation XML** : Si nous ajoutons le moindre caractère (même non visible comme des retours chariots) dans la partie **SignedInfo**, cela changera l'empreinte cryptographique dans **SignatureValue**
- **SHA256** (SignatureMethod)
- **PKCS#1 v1.5 Padding** (aka RSASSA-PKCS1-v1_5) (SignatureMethod)
- **RSA (2048)** (SignatureMethod)
- **Base64**

ECRITURE PSEUDO-ALGORITHMIQUE

```
SignatureValue = Base64 ( RSA-2048 ( Padding/PKCS#1.5 ( SHA256 ( C14N#withComments ( SignedInfo XML Block ) ) ) ) )
```

Vous voila avec votre SignatureValue, vous venez de signer votre premier pseudo-KDM (presque ;-)

A L'INTÉRIEUR DE KEYINFO

KeyInfo intègre toute la **chaîne des certificats publics RSA de l'encodeur** au format x509.

Voici un exemple d'un début de **KeyInfo** avec un seul **X509Data**, c'est notre premier certificat x509 de la chaîne :

```
<ds:KeyInfo>
    <ds:X509Data>
        <ds:X509IssuerSerial>
            <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>
        </ds:X509IssuerSerial>
        <ds:X509Certificate>MIIEejCCA2KgAwIBAgICAoMwDQYJKoZIhvcNAQELBQAwgYIx
ITAfBgNVBAoTGERDMi5TTVBURS5ET1JFTUlMQUJTLkNPTTEaMBgGA1UECxMRREMuRE9SRU1JTEFC
Uy5DT00xGjAYBgNVBAMTES5EQy5ETVMuREMyLlNNUFRFMSUwIwYDVQQuExwrTEx2dVlOTzRZQkpT
cDlKam1sdjhvaXBwelE9MB4XDTA3MDEwMTAwMDAwMFoXDTI1MTIzMTIzNTk1OVowgY4xITAfBgNV
BAoTGERDMi5TTVBURS5ET1JFTUlMQUJTLkNPTTEaMBgGA1UECxMRREMuRE9SRU1JTEFCUy5DT00x
JjAkBgNVBAMTHUNTLkRNU0pQMkstODAxMTkuREMuREMyLlNNUFRFMSUwIwYDVQQuExxTUZZU1Nx
V3dqZWZwcHFhc01KbWZkbVM2bEk9MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAvGKl
+zNPc1gqagM3K1JiThcaQW9u9MOYtTJMTVRixOfu5sGIOYAFf6FmSf5VneEGG9td/iN6GplI1e5u
pTvkyw1CTmiuAp3pqcqLhM8vyPntzloICTNxe9AIiuOnJQO1khLGUQ1pgD5Mvpfya/4Iiqnq8upR
84PoLnnsEKvoxeuXa1iLrCyDZykP7Ybn8U5muzdAqr9YRuFzFzbbq8Fcrz1uPhyKeK0i6+ZQVdfT
8xy4BuNGQcucimPGoImu+2yfNbEcfFQffDA5bGzR2/XDJTLEaqNwe/WXZyGokZtIRzmFJYGb4Zy+
pYGWIK5umuhjNFETufNI48mybNZtelpjIwIDAQABo4HrMIHoMAwGA1UdEwEB/wQCMAAwCwYDVR0P
BAQDAgSwMB0GA1UdDgQWBBRJAVhJKpbCN5+mmpqwwmZ92ZLqUjCBqwYDVR0jBIGjMIGggBT4su+5
g07hgElKn0mOaW/yiKmnNKGBhKSBgTB/MSEwHwYDVQQKExhEQzIuU01QVEUuRE9SRU1JTEFCUy5D
T00xGjAYBgNVBAsTEURDLkRPUkVNSUxBQlMuQ09NMRcwFQYDVQQDEw4uRE1TLkRDMi5TTVBURTEl
MCMGA1UELhMcUlEvNTNSbXVMc2J6Z2ZQWEdsUlltSnJ1d01zPYIBAjANBgkqhkiG9w0BAQsFAAOC
AQEAPBpZ6Tkg9TZg2LamLaOCPE6ingBvqmsQHR5V4b99sNU/EvdGL2eJjZFw4YLsxaEmPppIGFdM
IE+QJHrED5YLaJ9i1VvTbD1jNfrclW4Sbc6+mlQKFC6vOD5/go3hLYC8A5w92ea75iX77SuD0DCD
1GIs6FL3EGSrMXC59HFK4ahnIkxcUyhV75Cn14ujaOQpiO+zBxr/Vai/PDZoy6KSm0yYJEh97q4+
vQYQlRWSP1cF9IjLusrYBTjU30f7rygOR4vcqL8xnNQhYprBlJBn+KS3FqNKtrUoHTF0YtLl4I5l
90octl+vtQNGWxq4xU33Du9qXOBXBE9heSfG3CsmVw==</ds:X509Certificate>
    </ds:X509Data>
    <!--
        ...autres X509Data...
    -->
</ds:KeyInfo>
```

Selon la norme SMPTE, vous aurez au moins un **KeyInfo**, donc un certificat public. Dans les faits, dans la majorité des KDM, vous verrez au minimum **trois X509Data** à la suite, donc trois certificats publics :

- Le **certificat de l'encodeur**
- Le(s) **certificat(s) intermédiaire(s)**
- Le **certificat racine** (root certificate)

Le premier en haut de la pile est toujours le **certificat public de l'encodeur**, les suivants sont **ses parents**, surnommés certificats intermédiaires, il n'y a pas de limite dans leurs nombres, vous pouvez en avoir un seul comme des milliers (personne ne fait cela) et enfin le tout dernier est toujours le **certificat racine** qui chapeaute le tout.

Voyons tous les **IssuerName** dans **KeyInfo** de notre exemple :

```
<ds:X509IssuerName>dnQualifier=\+LLvuYNO4YBJSp9Jjmlv8oippzQ=,CN=.DC.DMS.DC2.SMPTE,OU=DC.DOREMILABS.COM,O=DC2.SMPTE.DOREMILABS.COM</ds:X509IssuerName>
<ds:X509IssuerName>dnQualifier=RQ/53RmuLsbzgfPXGlRYmJruwMs=,CN=.DMS.DC2.SMPTE,OU=DC.DOREMILABS.COM,O=DC2.SMPTE.DOREMILABS.COM</ds:X509IssuerName>
<ds:X509IssuerName>dnQualifier=pkCB9j5KrAjndhcBkc3fOdfL/BQ=,CN=.PRODUCTS.DC2.SMPTE,OU=DC.DOREMILABS.COM,O=DC2.SMPTE.DOREMILABS.COM</ds:X509IssuerName>
<ds:X509IssuerName>dnQualifier=a/wUIHLuFW7RKXpNQGmQ966xOv8=,CN=.ROOT.DC2.SMPTE,OU=DC.DOREMILABS.COM,O=DC2.SMPTE.DOREMILABS.COM</ds:X509IssuerName>
<ds:X509IssuerName>dnQualifier=a/wUIHLuFW7RKXpNQGmQ966xOv8=,CN=.ROOT.DC2.SMPTE,OU=DC.DOREMILABS.COM,O=DC2.SMPTE.DOREMILABS.COM</ds:X509IssuerName>
```

N'oubliez pas que **IssuerName** est le **SubjectName** du parent. Ainsi, il est normal que le dernier ait le même **IssuerName**, c'est un certificat signé par lui-même (self-signed).

Prenons le premier certificat et intégrons-le dans un container au format PEM :

```
-----BEGIN CERTIFICATE-----
MIIEejCCA2KgAwIBAgICAoMwDQYJKoZIhvcNAQELBQAwgYIxITAfBgNVBAoTGERD
Mi5TTVBURS5ET1JFTUlMQUJTLkNPTTEaMBgGA1UECxMRREMuRE9SRU1JTEFCUy5D
T00xGjAYBgNVBAMTES5EQy5ETVMuREMyLlNNUFRFMSUwIwYDVQQuExwrTEx2dVlO
TzRZQkpTcDlKam1sdjhvaXBwelE9MB4XDTA3MDEwMTAwMDAwMFoXDTI1MTIzMTIz
NTk1OVowgY4xITAfBgNVBAoTGERDMi5TTVBURS5ET1JFTUlMQUJTLkNPTTEaMBgG
A1UECxMRREMuRE9SRU1JTEFCUy5DT00xJjAkBgNVBAMTHUNTLkRNU0pQMkstODAx
MTkuREMuREMyLlNNUFRFMSUwIwYDVQQuExxTUUZZU1NxV3dqZWZwcHFhc01KbWZk
bVM2bEk9MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAvGKl+zNPc1gq
agM3K1JiThcaQW9u9MOYtTJMTVRixOfu5sGIOYAFf6FmSf5VneEGG9td/iN6GplI
1e5upTvkyw1CTmiuAp3pqcqLhM8vyPntzloICTNxe9AIiuOnJQO1khLGUQ1pgD5M
vpfya/4Iiqnq8upR84PoLnnsEKvoxeuXa1iLrCyDZykP7Ybn8U5muzdAqr9YRuFz
Fzbbq8Fcrz1uPhyKeK0i6+ZQVdfT8xy4BuNGQcucimPGoImu+2yfNbEcfFQffDA5
bGzR2/XDJTLEaqNwe/WXZyGokZtIRzmFJYGb4Zy+pYGWIK5umuhjNFETufNI48my
bNZtelpjIwIDAQABo4HrMIHoMAwGA1UdEwEB/wQCMAAwCwYDVR0PBAQDAgSwMB0G
A1UdDgQWBBRJAVhJKpbCN5+mmpqwwmZ92ZLqUjCBqwYDVR0jBIGjMIGggBT4su+5
g07hgElKn0mOaW/yiKmnNKGBhKSBgTB/MSEwHwYDVQQKExhEQzIuU01QVEUuRE9S
RU1JTEFCUy5DT00xGjAYBgNVBAsTEURDLkRPUkVNSUxBQlMuQ09NMRcwFQYDVQQD
Ew4uRE1TLkRDMi5TTVBURTElMCMGA1UELhMcUlEvNTNSbXVMc2J6Z2ZQWEdsUllt
SnJ1d01zPYIBAjANBgkqhkiG9w0BAQsFAAOCAQEAPBpZ6Tkg9TZg2LamLaOCPE6i
ngBvqmsQHR5V4b99sNU/EvdGL2eJjZFw4YLsxaEmPppIGFdMIE+QJHrED5YLaJ9i
1VvTbD1jNfrclW4Sbc6+mlQKFC6vOD5/go3hLYC8A5w92ea75iX77SuD0DCD1GIs
6FL3EGSrMXC59HFK4ahnIkxcUyhV75Cn14ujaOQpiO+zBxr/Vai/PDZoy6KSm0yY
JEh97q4+vQYQlRWSP1cF9IjLusrYBTjU30f7rygOR4vcqL8xnNQhYprBlJBn+KS3
FqNKtrUoHTF0YtLl4I5l90octl+vtQNGWxq4xU33Du9qXOBXBE9heSfG3CsmVw==
-----END CERTIFICATE-----
```

Puis, nous le faisons passer à OpenSSL pour analyse :

```
# openssl x509 -in cert.pem -text
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:
                    62:4e:17:1a:41:6f:6e:f4:c3:98:b5:32:4c:4d:54:
                    62:c4:e7:ee:e6:c1:88:39:80:05:7f:a1:66:49:fe:
                    55:9d:e1:06:1b:db:5d:fe:23:7a:1a:99:48:d5:ee:
                    6e:a5:3b:e4:cb:0d:42:4e:68:ae:02:9d:e9:a9:ca:
                    8b:84:cf:2f:c8:f9:ed:ce:5a:08:09:33:71:7b:d0:
                    08:8a:e3:a7:25:03:b5:92:12:c6:51:0d:69:80:3e:
                    4c:be:97:f2:6b:fe:08:8a:a9:ea:f2:ea:51:f3:83:
                    e8:2e:79:ec:10:ab:e8:c5:eb:97:6b:58:8b:ac:2c:
                    83:67:29:0f:ed:86:e7:f1:4e:66:bb:37:40:aa:bf:
                    58:46:e1:73:17:36:db:ab:c1:5c:af:3d:6e:3e:1c:
                    8a:78:ad:22:eb:e6:50:55:d7:d3:f3:1c:b8:06:e3:
                    46:41:cb:9c:8a:63:c6:a0:89:ae:fb:6c:9f:35:b1:
                    1c:7c:54:1f:7c:30:39:6c:6c:d1:db:f5:c3:25:32:
                    c4:6a:a3:70:7b:f5:97:67:21:a8:91:9b:48:47:39:
                    85:25:81:9b:e1:9c:be:a5:81:96:20:ae:6e:9a:e8:
                    63:34:51:13:b9:f3:48:e3:c9:b2:6c:d6:6d:7a:5a:
                    63:23
                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
    Signature Value:
        3c:1a:59:e9:39:20:f5:36:60:d8:b6:a6:2d:a3:82:3c:4e:a2:
        9e:00:6f:aa:6b:10:1d:1e:55:e1:bf:7d:b0:d5:3f:12:f7:46:
        2f:67:89:8d:91:70:e1:82:ec:c5:a1:26:3e:9a:48:18:57:4c:
        20:4f:90:24:7a:c4:0f:96:0b:68:9f:62:d5:5b:d3:6c:3d:63:
        35:fa:dc:95:6e:12:6d:ce:be:9a:54:0a:14:2e:af:38:3e:7f:
        82:8d:e1:2d:80:bc:03:9c:3d:d9:e6:bb:e6:25:fb:ed:2b:83:
        d0:30:83:d4:62:2c:e8:52:f7:10:64:ab:31:70:b9:f4:71:4a:
        e1:a8:67:22:4c:5c:53:28:55:ef:90:a7:d7:8b:a3:68:e4:29:
        88:ef:b3:07:1a:ff:55:a8:bf:3c:36:68:cb:a2:92:9b:4c:98:
        24:48:7d:ee:ae:3e:bd:06:10:95:15:92:3f:57:05:f4:88:cb:
        ba:ca:d8:05:38:d4:df:47:fb:af:28:0e:47:8b:dc:a8:bf:31:
        9c:d4:21:62:9a:c1:94:90:67:f8:a4:b7:16:a3:4a:b6:b5:28:
        1d:31:74:62:d2:e5:e0:8e:65:f7:4a:1c:b6:5f:af:b5:03:46:
        5b:1a:b8:c5:4d:f7:0e:ef:6a:5c:e0:57:04:4f:61:79:27:c6:
        dc:2b:26:57
```

Ceci est notre certificat public de notre encodeur, vous remarquerez que pour son **SubjectName** (avec le dnQualifier `SQFYSSqWwjefppqasMJmfdmS6lI=`) n'apparait nulle part dans le KDM, vous ne verrez que le **IssuerName** (avec le dnQualifier `+LLvuYNO4YBJSp9Jjmlv8oippzQ=`)

Analysons la suite des **X509Certificate** et ne récupérons que les **IssuerName** et **SubjectName** :

> Pour des raisons de lecture et compréhension, l'ordre des champs a été inversé en commençant par le dnQualifier.

```
               # 1er X509Certificate (encoder)
+-------------  Issuer  : dnQualifier = "+LLvuYNO4YBJSp9Jjmlv8oippzQ=", CN=.DC.DMS.DC2.SMPTE, OU=DC.DOREMILABS.COM, O=DC2.SMPTE.DOREMILABS.COM
|              Subject : dnQualifier = "SQFYSSqWwjefppqasMJmfdmS6lI=", CN=CS.DMSJP2K-80119.DC.DC2.SMPTE, OU=DC.DOREMILABS.COM, O=DC2.SMPTE.DOREMILABS.COM
|
|              # 2eme X509Certificate (dc)
|  +----------  Issuer  : dnQualifier = "RQ/53RmuLsbzgfPXGlRYmJruwMs=", CN=.DMS.DC2.SMPTE, OU=DC.DOREMILABS.COM, O=DC2.SMPTE.DOREMILABS.COM
+--|--------->  Subject : dnQualifier = "+LLvuYNO4YBJSp9Jjmlv8oippzQ=", CN=.DC.DMS.DC2.SMPTE, OU=DC.DOREMILABS.COM, O=DC2.SMPTE.DOREMILABS.COM
   |
   |           # 3eme X509Certificate (dms)
   |  +-------  Issuer  : dnQualifier = "pkCB9j5KrAjndhcBkc3fOdfL/BQ=", CN=.PRODUCTS.DC2.SMPTE, OU=DC.DOREMILABS.COM, O=DC2.SMPTE.DOREMILABS.COM
   +--|------>  Subject : dnQualifier = "RQ/53RmuLsbzgfPXGlRYmJruwMs=", CN=.DMS.DC2.SMPTE, OU=DC.DOREMILABS.COM, O=DC2.SMPTE.DOREMILABS.COM
      |
      |        # 4eme X509Certificate (product)
      |  +----  Issuer  : dnQualifier = "a/wUIHLuFW7RKXpNQGmQ966xOv8=", CN=.ROOT.DC2.SMPTE, OU=DC.DOREMILABS.COM, O=DC2.SMPTE.DOREMILABS.COM
      +--|--->  Subject : dnQualifier = "pkCB9j5KrAjndhcBkc3fOdfL/BQ=", CN=.PRODUCTS.DC2.SMPTE, OU=DC.DOREMILABS.COM, O=DC2.SMPTE.DOREMILABS.COM
         |
         |     # 5eme X509Certificate (root)
         |     Issuer  : dnQualifier = "a/wUIHLuFW7RKXpNQGmQ966xOv8=", CN=.ROOT.DC2.SMPTE, OU=DC.DOREMILABS.COM, O=DC2.SMPTE.DOREMILABS.COM
         +--->  Subject : dnQualifier = "a/wUIHLuFW7RKXpNQGmQ966xOv8=", CN=.ROOT.DC2.SMPTE, OU=DC.DOREMILABS.COM, O=DC2.SMPTE.DOREMILABS.COM
```

Vous vous demandez pourquoi nous intégrons l'ensemble de la chaîne de certificats dans chaque KDM : Toute cette succession de certificats permet à des récepteurs (players par exemple) de pouvoir vérifier la signature cryptographique d'un KDM.

Les players en salle ne sont pas censés avoir une connexion disponible vers Internet. S'ils n'ont pas les certificats publics de l'encodeur, et qu'ils ne peuvent pas les récupérer quelque part, ils ne pourront valider le KDM par eux-mêmes, et les players mis sur le marché ne peuvent contenir l'ensemble des certificats des encodeurs du monde entier.

Ainsi les KDM servent aussi de transporteur de certificats publics.

Vous trouverez plus d'infos à propos des certificats dans le chapitre Certificats.

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 des MXF, signer les PKL, signer les CPL, et signer les KDM, ...

NOTES

1. *« The **DigestValue** field of all the Reference elements shall be the Base64 encoded output of the SHA-256 hash of the referenced node. -- SMPTE 430-3 - Generic Extra-Theatre Message Format (ETM) »* ↩
2. *« The **SignatureValue** element shall be the output of the http://www.w3.org/2001/04/xmldsig-more#rsa-sha256 operation that is used to generate the signature. -- SMPTE 430-3 - Generic Extra-Theatre Message Format (ETM) »* et *« The SignatureMethod is the algorithm that is used to convert the canonicalized SignedInfo into the SignatureValue. »* -- RFC-3275 ↩
3. *« This implies the PKCS#1 v1.5 padding algorithm [RFC3447] as described in Section 2.3.1, but with the ASN.1 BER SHA-256 algorithm designator prefix »* --- RFC-6931 et *« 6.4.2 RSA (PKCS#1 v1.5), Identifier: http://www.w3.org/2001/04/xmldsig-more#rsa-sha256. (...) this specification refers to the RSASSA-PKCS1-v1_5 algorithm. »* -- xmldsig-core - PKCS1 et PKCS #1: RSA Cryptography Specifications ↩