

ASN.1:

Introduction

Zdeněk Říha





ASN.1

- Abstract Syntax Notation 1
- notation for describing abstract types and values
- Defined in ITU-T X.680 ... X.695
- Used in many file formats, including crypto
 - Public keys, private keys
 - Certificate requests, certificates
 - Digital signatures, padding, encrypted files



ASN.1

- Allows format/storage/transmission of data
 - Compatible among many applications
 - Not dependent on HW platform
 - E.g. little/big endian
 - Not dependent on operating system
- Simple & Structured types
- Multiple encoding rules (methods)

ASN.1 – Types



Type	Tag number (decimal)	Tag number (hexadecimal)
INTEGER	2	02
BIT STRING	3	03
OCTET STRING	4	04
NULL	5	05
OBJECT IDENTIFIER	6	06
SEQUENCE and SEQUENCE OF	16	10
SET and SET OF	17	11
PrintableString	19	13
IA5String	22	16
UTCTime	23	17





ASN.1 – simple types

- Integer
 - signed integer (there's no unsigned integer)
- Bit string
 - The number of bits does not have to be a multiple of 8
- Octet string
 - an arbitrary string of octets
- NULL
 - No data (used in parameters)
- PrintableString, IA5String, UTF8String, ...
 - Strings – the sets of characters are various
- UTCTime
 - Time



ASN.1 – OID type

- Object identifier (OID)
 - Sequence of integer components that identify an object
 - Assigned in a hierarchical way
- Example
 - sha-1WithRSAEncryption = 1.2.840.113549.1.1.5
 - iso(1) member-body(2)
us(840) rsadsi(113549)
pkcs(1) pkcs-1(1) 5
 - [1.2.840.113549.1.1](#) - PKCS-1
 - [1.2.840.113549.1](#) - PKCS
 - [1.2.840.113549](#) - RSADSI
 - [1.2.840](#) - USA
 - [1.2](#) - ISO member body
 - [1](#) - ISO assigned OIDs
 - [Top of OID tree](#)



ASN.1 – structured types

- SEQUENCE
 - an ordered collection of one or more types
- SEQUENCE OF
 - an ordered collection of zero or more occurrences of a given type
- SET
 - an unordered collection of one or more types
- SET OF
 - an unordered collection of zero or more occurrences of a given type



ASN.1 Encoding Rules

- XML – oriented formats
 - XER (XML Encoding Rules)
- Byte-oriented formats
 - BER (Basic Encoding Rules)
 - CER (Canonical Encoding Rules) – subset of BER
 - **DER (Distinguished Encoding Rules) – subset of BER**
 - **Used for crypto files**
- Bit-oriented formats
 - PER (Packed Encoding Rules)
- Verbose, human readable formats
 - GSER (Generic String Encoding Rules)



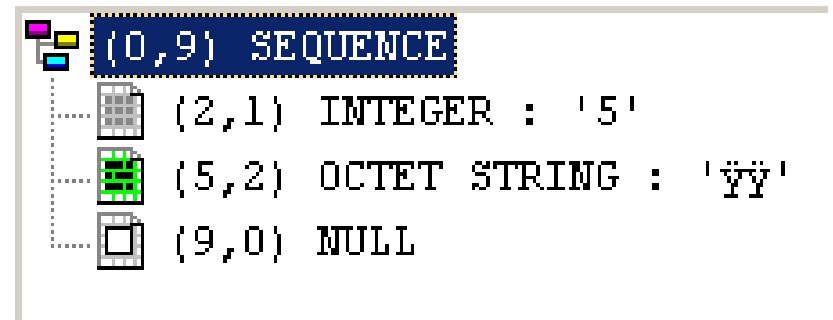
BER encoding

- TLV – Tag Length Value
 - All the data is encoded using a simple TLV format
 - Tag – what kind of data it is
 - Length – the length of the data
 - Value – the data itself
- Example
 - 02 01 05 [hexadecimal values]
 - Tag – Integer
 - Length of data – 1 byte
 - Data: (positive integer) 5



Nested data

- SEQUENCE is similar to struct/record
- 30 09 02 01 05 04 02 FF FF 05 00
 - 30 09 – sequence of length 9 bytes
 - 02 01 05 – integer 5
 - 04 02 FF FF – octet string FF FF
 - 05 00 – NULL (no data)





BER tags

- Tag encoding



- Class

Class	Bit 8	Bit 7
universal	0	0
application	0	1
context-specific	1	0
private	1	1

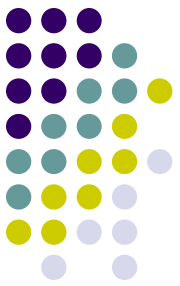
- Tag number

- Bits 1-5
- If all bits are 1 then the tag continues in the following byte(s)



BER length

- length ≥ 0 && length ≤ 127
 - The length is coded directly
 - E.g. '05'
- Otherwise the bit 8 is set, bits 1-7 code the number of bytes that specify the length
 - E.g. 255 -> '81' 'FF'
 - E.g. 256 -> '82' '01' '00' or also '83' '00' '01' '00'
 - BER x DER
- '80' is “indefinite” length
 - Not allowed in DER



BER value

- The data itself
- Dependent on data type
 - Integer: signed – e.g. 128 -> '00 80'
 - Octet string: directly the data
 - Bit string: number of unused bits + padded bit string to a multiple of 8 bits (padding is at the end)
 - UTCTime: string of one of the forms

YYMMDDhhmmZ

YYMMDDhhmm+hh'mm'

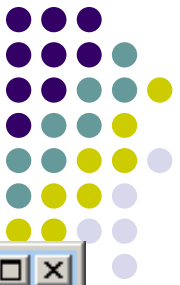
YYMMDDhhmm-hh'mm'

YYMMDDhhmmssZ

YYMMDDhhmmss+hh'mm'

YYMMDDhhmmss-hh'mm'

First look at the binary DER file



```
TSL_1.cer (~\Plocha\PKI) - GVIM
Soubor Úpravy Nástroje Syntaxe Buffery Okna Nápověda
[Icons]
G, ^DL0, ^C* ^C^B^A^B^B^C^W&E0^M^F *†H†÷^M^A^A^K^E^@0M<<1^K0 ^F^CU^D^F^S^BCZ1907
^F^CU^D^C^L0I.CA - Standard Certification Authority, 09/20091-0+^F^CU^D
^L$Prvně- certifikační- autorita, a.s.1200^F^CU^D^K^L)I.CA - Provider of Certification Ser
vices0^^W^M091221173259Z^W^M101221173259Z0M-1^K0 ^F^CU^D^F^S^BCZ1705^F^CU^D^C^L.Ministr
y of the Interior of the Czech Republic1705^F^CU^D
^L.Ministry of the Interior of the Czech Republic1^U0^S^F^CU^D^E^S^LICA - 6139660, ^A"0^M^F
*†H†÷^M^A^A^A^E^@^C, ^A^0^@0, ^A
^B, ^A^A^@^ jHŠ05T0é0 Ě°/Z, lxA}^T^e[...Ě'x\šUšĚĚ»R'ó%č|f4çL^^LôóÁ%-ëIá0č^PĪĪ'-J'¿Jç%]U°"
^_ j]Ěö=Í$}WÁ^ádBAžž2Ú''Ōr^Z††;uď^PBRD%P^OŮp, MĚ^\<ž^M^\ZdýóÁn+3Mó6P^UUpw,-$cĪä+Gvó0Q^R
UKLF~±g^R^B(/EĐšACšQ,ĚŌH [ĀUÝĐ^G{ó^WĀĀg0•ä^E^M<wTĚHb\CLó7^Kúš0F^V^]q^GžžGLv"
>>^Y^C^W~ě.'-“ť
č■ă^TDó^0 ^T^X0^DŮž],l}Ň"> tu^B^C^A^@^A&■Ů0■ř0^Q^F^CU^]%^D
0^H^F^F^D^@^7^C^@0^_F `†H^A†řB^A^M^D^R^U^P92030300000112730^_F^CU^]#^D^X0^U^M^TÁL8"Ō†HŮ"
■,óĪ^Y^PÚg#0^]F^CU^]N^D^U^D^Tíu&\$IjŇ#P(LĀLą)>[ŮĚB0^Z^F^CU^] ^D^S0^Q0^D^F^M+^F^A^D^A■,H^A
^A<C^@0^K^F^CU^]0^D^D^C^B^G■0Y^F^CU^] ^_DR0P0& $ "† http://scr1dp1.ica.cz/sica09.cr10& $
"† http://scr1dp2.ica.cz/sica09.cr10^M^F *†H†÷^M^A^A^K^E^@^C, ^A^A^@a#Mó^UrĚJŮĚĚ--^SE^
C~Ī^Ā60d*F~ž"‰5Ř_ ^Súyč'ōĎIáěâ^OüyR^By^Y^L6D-6^S'^Lž,,ŇŇ0v^_š■7%, "žg•v wĚĚ^K-šđrž
e_ ^ZlxžĪkš%š^Ear^0& ě5"ř8Nt■^YŮ■>>%[^W7ÉN00&t"†x
ř
á^MÍ^XúkĚŇ#ăă^W^@~ ±4â@T0■^A4'ř±^D†J8Ě, 0Ř*×šâGPoAXj^Ažm0IPŮX0Rš$-:ćĀ^Gü^Xq>K'<A;■d{Ů0Ā8÷i†J8$@
±^BĚ0>?"ŏš~ć-{Mđ'úáF<'Ň^UŇ&ě-■RŌ"'+ĪĪ•Y^Cwd^SW
~
~
1,1 Uše
```



DER vs. PEM

- PEM
 - Privacy Enhanced Mail
- PEM as such not used, but formats still used
- Textual formats
 - Practical for transport channels where full 8bit data can be damaged
- PEM is base64 coded DER enveloped with
 - -----BEGIN **SOMETHING**-----
 - -----END **SOMETHING**-----
 - Where **SOMETHING** is CERTIFICATE/PKCS7/KEY...

Sample PEM file



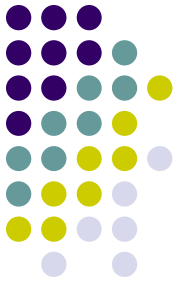
```
cscsa.pem (~\Plocha\Vyuka_p...1\PV181_drive\ASN.1) - GYIM1
Soubor Úpravy Nástroje Syntaxe Buffery Okna Nápověda
-----BEGIN CERTIFICATE-----
MIIE8jCCAyagAwIBAgIBATBBBgkqhkiG9w0BAQowNKAPMA0GCWCGSAF1AwQCAQUA
oRwwGgYJKoZIhvcNAQEIOMA0GCWCGSAF1AwQCAQUAogMCASAwUzELMAkGA1UEBhMC
Q1oxFzAUBGNUBAOTDkN6ZWNoIFJlcHUibG1jMR0wGwYDUQLEXRNaW5pc3RyeSBv
ZiBjbjRlcm1vcjEQAQA4GA1UEAxQHQ1NDQU9DWjAeFw0wNjA3MjQwMDAwMDBaFw0y
MTEwMjQyMzU5NTIaMFcxZjA3BGNUBAYTAkNAwMRcW50ZDQ0YDUQKEw5DemUjaCBSZXB1
YmxpYzEdMBsGA1UECXMUTWluaXN0cnkgb2YgSW50ZXJpb3IxEDA0BgNUBAMUB0NT
Q0FFQ1owggGiMA0GCsGqS1b3DQEBQUAA4IBjwAwggGKAoIBgQCcuUznqqCTF+LC1
aqULMoUsigvNh0cqWFKu+XGy4NuS3Je0LICgRZe9A3IUf5N0ArDN3jdmJrX1ug0
0XwuRgG+800ifmMH32kFLyB0+RbPFm0JWi3v7mxwMdtLQw1xTdhgu/WMPPrxn1bf
Qm3I2XhwTvrBs2mI6q1y54ibm0c63UsAZdqDc+t9AIX11oFwq3z04MBxMkCYsEfH
Joy1B9UhuFdk5pGEdTWUTs8aRuPFWrS3WzhSmoWDiR8hCiZnYhSjx5I8g/vKFRyj
JtpJXaqrWRbnfNL+iSJ15cCUH9f+bIL026B2Y6tF8EsNiIoay/qewEKA1NdxXczJ
190ShkUuKeUrpY1Uhd/B9g6vXUMrkznax51273KS79kk8GgcwZmY87q2wp1wE/Q6
Rc/iD14Bcum/nezXUrb+vnMprbSwid7Wt7e5z2rXtsP/56Sa01N/kJ3C+UK1Suhd
9kT0vmlPUNwOUK1d75WqRKZb6B+JtNuBCeyu89wrGkt527RF3kCAwEAANhMF8w
HQYDUR00BBYEFLSBmFXskNo/DW+f0n3n4MF11JYsMA4GA1UdDwEB/wQEAwIBBjAa
BgNUHSAAEzARMA8GDSqBS7cYAQEBAYN1smswEgYDUR0TAQH/BAGwBgEB/wIBADBB
BgkqhkiG9w0BAQowNKAPMA0GCWCGSAF1AwQCAQUAogMCASADggGBAChyozpMnqq+HarcDKatzMbFnbG4YlgbZXF5
kUsAK3y8qWli1oVI6TW8U199xsR/GUACjJ1YLE8hiHjmtG8mSh8MUM7qqf0JnjFo
3g5/q/jJH7+d6BnPGWsc0s/vwzfla10a/bozYe0Yq9drMkdZTF0GNEDWisWma4RQ
B5F7ithB+/7dxn23x0rJcoemkw4qeCbZn86FToMo2eNc8Cbt1I6AixDzzKC67LS8
Yi0b0FwPn5U09aBwcW5oUUGUmeeq9XRb7nkocHm6E1pW1hwFVeJfQR0hDSKazf
eFrRYPb7n2MsAg1wLHAB0JPOeA7yENjXh5maybtv+ksUfdJ469f4n4cvUyQ0eDtZ
XBDmG2Y0UyaS0jxVkhStBR2PTW1s9cvL2wxf/6Nnq9gpzIf+UzBJSxGyrwDwKkNA
tnFnFsk3q93/7t0qmIyf2sxCi95CFTF1R2Br55GwqCczFT5DzHt4NKXWiaX0DFC+
6MTSBSW50/G5ZryNPNI79qLqhXn+Q==
-----END CERTIFICATE-----
22,64 Uše
```




ASN.1 viewers

- Unber (part of asn1c)
- Openssl asn1parse
- ASN.1 Editor
- ...

OpenSSL asn1parse



```
C:\WINDOWS\system32\cmd.exe
C:\Program Files\OpenSSL\bin>openssl.exe asn1parse -inform DER -in CZE_CSCA_2009
0113.der
 0:d=0  hl=4  l=1266  cons: SEQUENCE
 4:d=1  hl=4  l= 806  cons: SEQUENCE
 8:d=2  hl=2  l=   3  cons: cont [ 0 ]
10:d=3  hl=2  l=   1  prim: INTEGER           :02
13:d=2  hl=2  l=   1  prim: INTEGER           :3A
16:d=2  hl=2  l=  65  cons: SEQUENCE
18:d=3  hl=2  l=   9  prim: OBJECT            :1.2.840.113549.1.1.10
29:d=3  hl=2  l=  52  cons: SEQUENCE
31:d=4  hl=2  l=  15  cons: cont [ 0 ]
33:d=5  hl=2  l=  13  cons: SEQUENCE
35:d=6  hl=2  l=   9  prim: OBJECT            :sha256
46:d=6  hl=2  l=   0  prim: NULL
48:d=4  hl=2  l=  28  cons: cont [ 1 ]
50:d=5  hl=2  l=  26  cons: SEQUENCE
52:d=6  hl=2  l=   9  prim: OBJECT            :1.2.840.113549.1.1.8
63:d=6  hl=2  l=  13  cons: SEQUENCE
65:d=7  hl=2  l=   9  prim: OBJECT            :sha256
76:d=7  hl=2  l=   0  prim: NULL
78:d=4  hl=2  l=   3  cons: cont [ 2 ]
80:d=5  hl=2  l=   1  prim: INTEGER           :20
83:d=2  hl=2  l=  87  cons: SEQUENCE
85:d=3  hl=2  l=  11  cons: SET
87:d=4  hl=2  l=   9  cons: SEQUENCE
89:d=5  hl=2  l=   3  prim: OBJECT            :countryName
94:d=5  hl=2  l=   2  prim: PRINTABLESTRING  :CZ
98:d=3  hl=2  l=  23  cons: SET
100:d=4  hl=2  l=  21  cons: SEQUENCE
102:d=5  hl=2  l=   3  prim: OBJECT            :organizationName
107:d=5  hl=2  l=  14  prim: PRINTABLESTRING  :Czech Republic
123:d=3  hl=2  l=  29  cons: SET
125:d=4  hl=2  l=  27  cons: SEQUENCE
127:d=5  hl=2  l=   3  prim: OBJECT            :organizationalUnitName
132:d=5  hl=2  l=  20  prim: PRINTABLESTRING  :Ministry of Interior
154:d=3  hl=2  l=  16  cons: SET
156:d=4  hl=2  l=  14  cons: SEQUENCE
158:d=5  hl=2  l=   3  prim: OBJECT            :commonName
163:d=5  hl=2  l=   7  prim: T61STRING          :CSCA_CZ
172:d=2  hl=2  l=  30  cons: SEQUENCE
174:d=3  hl=2  l=  13  prim: UTCTIME             :090113000000Z
189:d=3  hl=2  l=  13  prim: UTCTIME             :240413000000Z
204:d=2  hl=2  l=  87  cons: SEQUENCE
206:d=3  hl=2  l=  11  cons: SET
208:d=4  hl=2  l=   9  cons: SEQUENCE
210:d=5  hl=2  l=   3  prim: OBJECT            :countryName
215:d=5  hl=2  l=   2  prim: PRINTABLESTRING  :CZ
219:d=3  hl=2  l=  23  cons: SET
221:d=4  hl=2  l=  21  cons: SEQUENCE
223:d=5  hl=2  l=   3  prim: OBJECT            :organizationName
228:d=5  hl=2  l=  14  prim: PRINTABLESTRING  :Czech Republic
244:d=3  hl=2  l=  29  cons: SET
246:d=4  hl=2  l=  27  cons: SEQUENCE
```

unber

CSCA_CZE.crt



```
anxur.fi.muni.cz - PuTTY
labak:~$ unber /usr/share/doc/dirmngr-1.0.3/examples/extra-certs/bnetza-10r-ocsp-2.crt
<C 0="0" T="[UNIVERSAL 16]" TL="4" V="952" A="SEQUENCE">
  <C 0="4" T="[UNIVERSAL 16]" TL="4" V="804" A="SEQUENCE">
    <C 0="8" T="[0]" TL="2" V="3">
      <P 0="10" T="[UNIVERSAL 2]" TL="2" V="1" A="INTEGER" F>2</P>
    </C 0="13" T="[0]" L="5">
      <P 0="13" T="[UNIVERSAL 2]" TL="2" V="1" A="INTEGER" F>53</P>
    <C 0="16" T="[UNIVERSAL 16]" TL="2" V="10" A="SEQUENCE">
      <P 0="18" T="[UNIVERSAL 6]" TL="2" V="6" A="OBJECT IDENTIFIER" F>1.3.36.3.3.1.2</P>
      <P 0="26" T="[UNIVERSAL 5]" TL="2" V="0" A="NULL"></P>
    </C 0="28" T="[UNIVERSAL 16]" A="SEQUENCE" L="12">
      <C 0="28" T="[UNIVERSAL 16]" TL="2" V="63" A="SEQUENCE">
        <C 0="30" T="[UNIVERSAL 17]" TL="2" V="11" A="SET">
          <C 0="32" T="[UNIVERSAL 16]" TL="2" V="9" A="SEQUENCE">
            <P 0="34" T="[UNIVERSAL 6]" TL="2" V="3" A="OBJECT IDENTIFIER" F>2.5.4.6</P>
            <P 0="39" T="[UNIVERSAL 19]" TL="2" V="2" A="PrintableString">DE</P>
          </C 0="43" T="[UNIVERSAL 16]" A="SEQUENCE" L="11">
            <C 0="43" T="[UNIVERSAL 17]" A="SET" L="13">
              <C 0="43" T="[UNIVERSAL 17]" TL="2" V="26" A="SET">
                <C 0="45" T="[UNIVERSAL 16]" TL="2" V="24" A="SEQUENCE">
                  <P 0="47" T="[UNIVERSAL 6]" TL="2" V="3" A="OBJECT IDENTIFIER" F>2.5.4.10</P>
                  <P 0="52" T="[UNIVERSAL 12]" TL="2" V="17" A="UTF8String">Bundesnetzagentur</P>
                </C 0="71" T="[UNIVERSAL 16]" A="SEQUENCE" L="26">
                  <C 0="71" T="[UNIVERSAL 17]" A="SET" L="28">
                    <C 0="71" T="[UNIVERSAL 17]" TL="2" V="20" A="SET">
                      <C 0="73" T="[UNIVERSAL 16]" TL="2" V="18" A="SEQUENCE">
                        <P 0="75" T="[UNIVERSAL 6]" TL="2" V="3" A="OBJECT IDENTIFIER" F>2.5.4.3</P>
                        <P 0="80" T="[UNIVERSAL 12]" TL="2" V="11" A="UTF8String">10R-CA 1:PN</P>
                      </C 0="93" T="[UNIVERSAL 16]" A="SEQUENCE" L="20">
                        <C 0="93" T="[UNIVERSAL 17]" A="SET" L="22">
                          </C 0="93" T="[UNIVERSAL 16]" A="SEQUENCE" L="65">
                            <C 0="93" T="[UNIVERSAL 16]" TL="2" V="30" A="SEQUENCE">
                              <P 0="95" T="[UNIVERSAL 23]" TL="2" V="13" A="UTCTime">0508040827092</P>
                              <P 0="110" T="[UNIVERSAL 23]" TL="2" V="13" A="UTCTime">0712310823492</P>
                            </C 0="125" T="[UNIVERSAL 16]" A="SEQUENCE" L="32">
                              <C 0="125" T="[UNIVERSAL 16]" TL="2" V="65" A="SEQUENCE">
                                <C 0="127" T="[UNIVERSAL 17]" TL="2" V="11" A="SET">
                                  <C 0="129" T="[UNIVERSAL 16]" TL="2" V="9" A="SEQUENCE">
                                    <P 0="131" T="[UNIVERSAL 6]" TL="2" V="3" A="OBJECT IDENTIFIER" F>2.5.4.6</P>
                                    <P 0="136" T="[UNIVERSAL 19]" TL="2" V="2" A="PrintableString">DE</P>
                                  </C 0="140" T="[UNIVERSAL 16]" A="SEQUENCE" L="11">
                                    </C 0="140" T="[UNIVERSAL 17]" A="SET" L="13">
                                      <C 0="140" T="[UNIVERSAL 17]" TL="2" V="26" A="SET">
                                        <C 0="142" T="[UNIVERSAL 16]" TL="2" V="24" A="SEQUENCE">
                                          <P 0="144" T="[UNIVERSAL 6]" TL="2" V="3" A="OBJECT IDENTIFIER" F>2.5.4.10</P>
                                          <P 0="149" T="[UNIVERSAL 12]" TL="2" V="17" A="UTF8String">Bundesnetzagentur</P>
                                        </C 0="168" T="[UNIVERSAL 16]" A="SEQUENCE" L="26">
```

Manual viewing/processing



```
Kcze_csca_20060724.cer (~\Plocha\PKI) - GVIM3
Soubor Úpravy Nástroje Syntaxe Buffery Okna Nápověda
[Icons]
0000000: 3082 04f2 3082 0326 a003 0201 0202 0101 0...0...&.....
0000010: 3041 0609 2a86 4886 f70d 0101 0a30 34a0 0A...*.H.....04.
0000020: 0f30 0d06 0960 8648 0165 0304 0201 0500 .0...`.H.e.....
0000030: a11c 301a 0609 2a86 4886 f70d 0101 0830 ..0...*.H.....0
0000040: 0d06 0960 8648 0165 0304 0201 0500 a203 ...`.H.e.....
0000050: 0201 2030 5731 0b30 0906 0355 0406 1302 ..0W1.0...U....
0000060: 435a 3117 3015 0603 5504 0a13 0e43 7a65 CZ1.0...U....Cze
0000070: 6368 2052 6570 7562 6c69 6331 1d30 1b06 ch Republic1.0..
0000080: 0355 040b 1314 4d69 6e69 7374 7279 206f .U....Ministry o
0000090: 6620 496e 7465 7269 6f72 3110 300e 0603 f Interior1.0...
00000a0: 5504 0314 0743 5343 415f 435a 301e 170d U....CSCA_CZ0...
00000b0: 3036 3037 3234 3030 3030 3030 5a17 0d32 060724000000Z..2
00000c0: 3131 3032 3432 3335 3935 395a 3057 310b 11024235959Z0W1.
00000d0: 3009 0603 5504 0613 0243 5a31 1730 1506 0...U....CZ1.0..
00000e0: 0355 040a 130e 437a 6563 6820 5265 7075 .U....Czech Repu
00000f0: 626c 6963 311d 301b 0603 5504 0b13 144d blic1.0...U....M
0000100: 696e 6973 7472 7920 6f66 2049 6e74 6572 inistry of Inter
0000110: 696f 7231 1030 0e06 0355 0403 1407 4353 ior1.0...U....CS
0000120: 4341 5f43 5a30 8201 a230 0d06 092a 8648 CA_CZ0...0...*.H
0000130: 86f7 0d01 0101 0500 0382 018f 0030 8201 .....0..
0000140: 8a02 8201 8100 af51 99ea a824 c5f8 b0b5 .....Q...$.
0000150: 6aa5 4b32 852c 8a0b cd84 e72a 59f2 aef9 j.K2.,.....*Y...
0000160: 71b2 e0db d2dc 97b4 2c80 a045 97bd 0372 q.....,E...r
0000170: 149d fe4d d00a c337 78dd 989a d7d6 e834 ...H...7x.....4
Počet filtrovaných řádků: 7
1,1 Začátek
```

- 30 82 04 f2
 - SEQUENCE
 - length 1266B
- 30 82 03 26
 - SEQUENCE
 - length 806B
- A0 03
 - CONTEXT SPECIFIC 0
 - Length 3B
- 02 01 02
 - INTEGER: 2



ASN.1 Grammar

- To understand the structure (what is the meaning of particular fields) we need ASN.1 grammar

```
CertificateList ::= SEQUENCE {  
  tbsCertList      TBSCertList,  
  signatureAlgorithm AlgorithmIdentifier,  
  signatureValue   BIT STRING }
```

```
TBSCertList ::= SEQUENCE {  
  version          Version OPTIONAL,  
                  -- if present, MUST be v2  
  signature        AlgorithmIdentifier,  
  issuer           Name,  
  thisUpdate      Time,  
  nextUpdate      Time OPTIONAL,  
  revokedCertificates SEQUENCE OF SEQUENCE {  
    userCertificate CertificateSerialNumber,  
    revocationDate  Time,  
    crlEntryExtensions Extensions OPTIONAL  
                  -- if present, MUST be v2  
  } OPTIONAL,  
  crlExtensions   [0] EXPLICIT Extensions OPTIONAL  
                  -- if present, MUST be v2  
}
```

ASN.1 – RSA keys

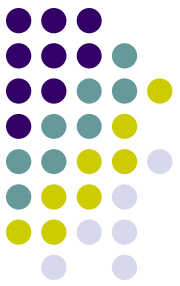


```
RSAPublicKey ::= SEQUENCE {
    modulus          INTEGER, -- n
    publicExponent  INTEGER -- e
}

--
-- Representation of RSA private key with information for the CRT algorithm.
--
RSAPrivateKey ::= SEQUENCE {
    version          Version,
    modulus          INTEGER, -- n
    publicExponent  INTEGER, -- e
    privateExponent INTEGER, -- d
    prime1          INTEGER, -- p
    prime2          INTEGER, -- q
    exponent1       INTEGER, -- d mod (p-1)
    exponent2       INTEGER, -- d mod (q-1)
    coefficient      INTEGER, -- (inverse of q) mod p
    otherPrimeInfos OtherPrimeInfos OPTIONAL
}
```

 RSA.key

Source:
PKCS#1



ASN.1 – RSA padding

- PKCS#1 v1.5

- $m = 0x00 \parallel 0x01 \parallel 0xFF \dots 0xFF \parallel 0x00 \parallel T$
- Where T is defined as DER encoding of

```
DigestInfo ::= SEQUENCE {  
    digestAlgorithm AlgorithmIdentifier,  
    digest OCTET STRING  
}
```

- In practice:

```
MD2:      (0x)30 20 30 0c 06 08 2a 86 48 86 f7 0d 02 02 05 00 04 10 || H.  
MD5:      (0x)30 20 30 0c 06 08 2a 86 48 86 f7 0d 02 05 05 00 04 10 || H.  
SHA-1:    (0x)30 21 30 09 06 05 2b 0e 03 02 1a 05 00 04 14 || H.  
SHA-256:  (0x)30 31 30 0d 06 09 60 86 48 01 65 03 04 02 01 05 00 04 20 || H.  
SHA-384:  (0x)30 41 30 0d 06 09 60 86 48 01 65 03 04 02 02 05 00 04 30 || H.  
SHA-512:  (0x)30 51 30 0d 06 09 60 86 48 01 65 03 04 02 03 05 00 04 40 || H.
```




ASN.1 – RSA signature

- RSA signature is the number $s = m^d \bmod n$

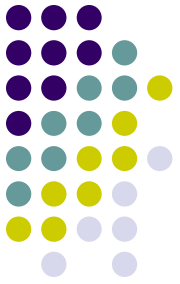
```
ASN.1 Editor - Opening File: postsignature_tsa_tsu1.der
File View Tools Help
(0,1818) SEQUENCE
+ (4,1538) SEQUENCE
- (1546,13) SEQUENCE
  (1548,9) OBJECT IDENTIFIER : : '1.2.840.113549.1.1.11'
  (1559,0) NULL
  (1561,257) BIT STRING UnusedBits: 0 : '7BA3DA2079DA32BC74B858B5ED5028EC4880D631D09B1758A1304491DBF5DE6A'
File Name: C:\Documents and Settings\Administrator\Plocha\PKI\postsignature_tsa_tsu1.der
Size: 1822 (bytes)
```

ASN.1 – signature OIDs



RSA Encryption ¹	1.2.840.113549.1.1.1
RSASSA-PKCS1_v15 with SHA1	1.2.840.113549.1.1.5
RSASSA-PSS	1.2.840.113549.1.1.10 (PKCS #1 Version 2.1)
RSASSA-PKCS1_v15 with SHA224	1.2.840.113549.1.1.14
RSASSA-PKCS1_v15 with SHA256	1.2.840.113549.1.1.11
RSASSA-PKCS1_v15 with SHA384	1.2.840.113549.1.1.12
RSASSA-PKCS1_v15 with SHA512	1.2.840.113549.1.1.13

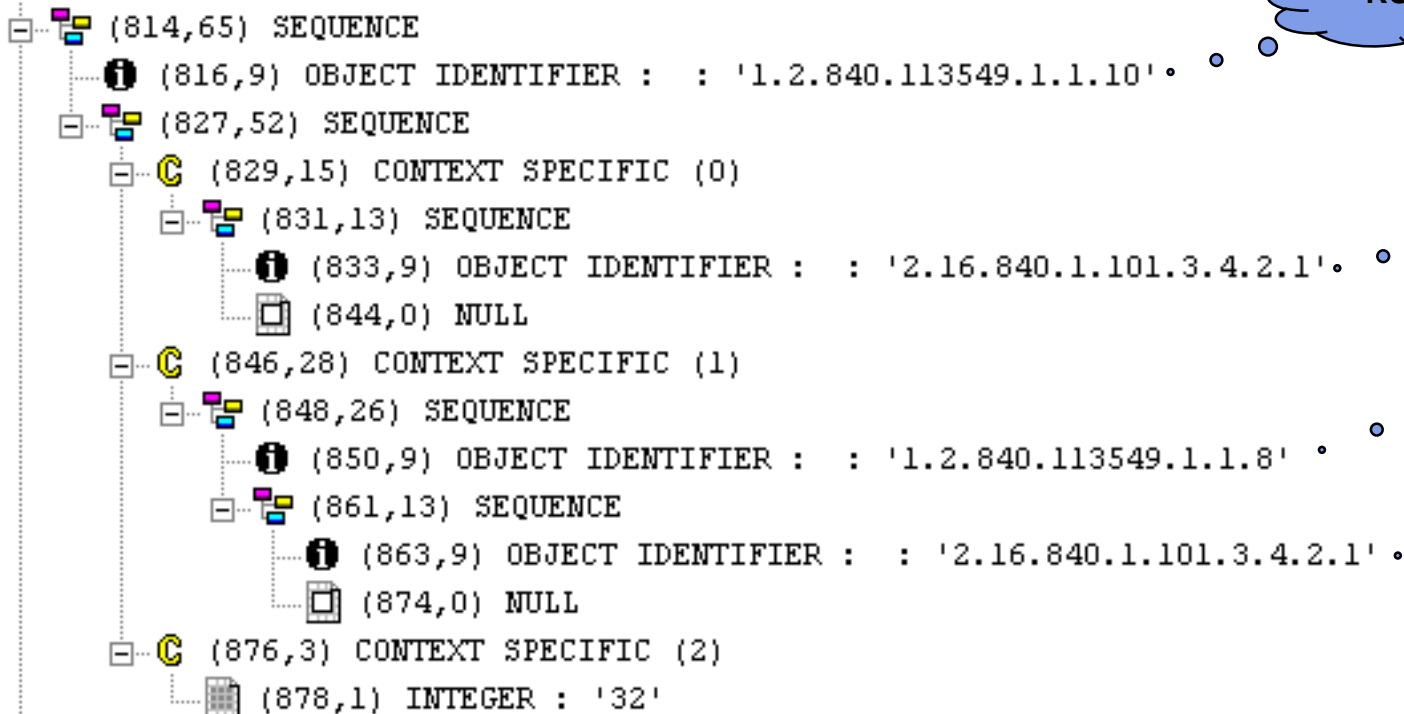
ASN.1 – RSA PSS params



```
RSASSA-PSS-params ::= SEQUENCE {  
    hashAlgorithm      [0] HashAlgorithm      DEFAULT sha1,  
    maskGenAlgorithm   [1] MaskGenAlgorithm   DEFAULT mgf1SHA1,  
    saltLength         [2] INTEGER           DEFAULT 20,  
    trailerField       [3] TrailerField       DEFAULT trailerFieldBC  
}
```

Source:
PKCS#1

```
TrailerField ::= INTEGER { trailerFieldBC(1) }
```



RSASSA-PSS

SHA256

MGF1

SHA256



ASN.1 – DSA keys

```
Dsa-Parms ::= SEQUENCE {  
    p      INTEGER,  
    q      INTEGER,  
    g      INTEGER }
```

Source:
RFC 5480

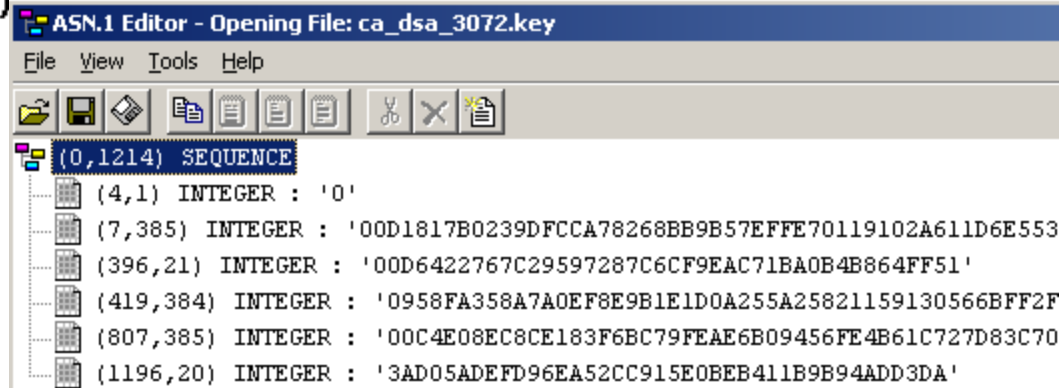
```
DSAPublicKey ::= INTEGER -- public key, Y
```

DSAPrivateKey is an INTEGER, usually denoted as X

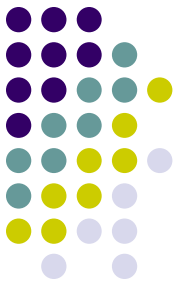
```
ASN1_SEQUENCE_cb(DSAPrivateKey, dsa_cb) = {  
    ASN1_SIMPLE(DSA, version, LONG),  
    ASN1_SIMPLE(DSA, p, BIGNUM),  
    ASN1_SIMPLE(DSA, q, BIGNUM),  
    ASN1_SIMPLE(DSA, g, BIGNUM),  
    ASN1_SIMPLE(DSA, pub_key, BIGNUM),  
    ASN1_SIMPLE(DSA, priv_key, BIGNUM)  
} ASN1_SEQUENCE_END_cb(DSA, DSAPrivateKey)
```

Source:
OpenSSL

 DSA.key



ASN.1 – DSA signature



```
Dss-Sig-Value ::= SEQUENCE {  
    r      INTEGER,  
    s      INTEGER }
```

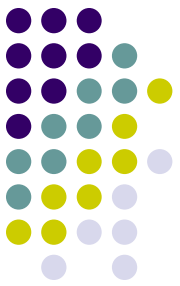
Source:
RFC 5480

The screenshot shows the ASN.1 Editor interface for the file 'ca_dsa_3072_sha1.crt'. The main window displays a tree view of the ASN.1 structure:

- (0,1681) SEQUENCE (highlighted)
 - (4,1617) SEQUENCE
 - (1625,9) SEQUENCE
 - (1627,7) OBJECT IDENTIFIER : dsaWithSha1 : '1.2.840.10040.4.3'
 - (1636,47) BIT STRING UnusedBits: 0
 - (1639,44) SEQUENCE
 - (1641,20) INTEGER : '64CA41FEA8CBA7E9282D215BC60BF4FECD198858'
 - (1663,20) INTEGER : '1B78E8B76423099D9D897F59066A813E93C3A7A1'

The status bar at the bottom indicates: File Name: C:\Documents and Settings\Administrator\Plocha\PKI\gen_sod\keys\ca_dsa Size: 1685 (bytes)

ASN.1 – DSA - OIDs



```
-- DSA with SHA-1
-- Parameters are ABSENT

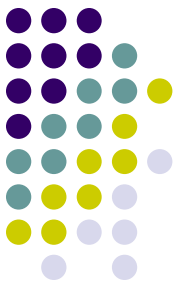
id-dsa-with-sha1 OBJECT IDENTIFIER ::= {
    iso(1) member-body(2) us(840) x9-57(10040) x9algorithm(4) 3 }

-- DSA with SHA-224
-- Parameters are ABSENT

id-dsa-with-sha224 OBJECT IDENTIFIER ::= {
    joint-iso-ccitt(2) country(16) us(840) organization(1) gov(101)
    csor(3) algorithms(4) id-dsa-with-sha2(3) 1 }

-- DSA with SHA-256
-- Parameters are ABSENT

id-dsa-with-sha256 OBJECT IDENTIFIER ::= {
    joint-iso-ccitt(2) country(16) us(840) organization(1) gov(101)
    csor(3) algorithms(4) id-dsa-with-sha2(3) 2 }
```



ASN.1 – ECDSA keys

```
ECParameters ::= SEQUENCE {  
    version      INTEGER{ecpVer1(1)} (ecpVer1),  
    fieldID      FieldID{{FieldTypes}},  
    curve        Curve,  
    base         ECPoint,  
    order        INTEGER,  
    cofactor    INTEGER OPTIONAL,  
    ...  
}
```

```
Curve ::= SEQUENCE {  
    a      FieldElement,  
    b      FieldElement,  
    seed   BIT STRING OPTIONAL  
}
```

```
ECPoint ::= OCTET STRING -- Elliptic curve point
```

```
ECPrivateKey{CURVES:IOSet} ::= SEQUENCE {  
    version      INTEGER { ecPrivkeyVer1(1) } ( ecPrivkeyVer1 ),  
    privateKey   OCTET STRING, • • •  
    parameters   [0] Parameters{{IOSet}} OPTIONAL, INTEGER  
    publicKey    [1] BIT STRING OPTIONAL  
}
```

```
SubjectPublicKeyInfo ::= SEQUENCE {  
    algorithm      AlgorithmIdentifier{{ECPKAlgorithms}},  
    subjectPublicKey BIT STRING • • • ECPoint  
}
```




ASN.1 – ECDSA signatures

```
ec-signature-value ::= SEQUENCE {  
    r    INTEGER,  
    s    INTEGER  
}
```

Source:
RFC 5480

1.2.840.10045.4.1 - ecdsa-with-SHA1

ASN.1 Editor - Opening File: Switzerland.crt

File View Tools Help

(0,1059) SEQUENCE

- (4,938) SEQUENCE
 - (946,9) SEQUENCE
 - (948,7) OBJECT IDENTIFIER : : '1.2.840.10045.4.1'**
 - (957,104) BIT STRING UnusedBits: 0
 - (960,101) SEQUENCE
 - (962,49) INTEGER : '00FEEB445183C58A9055C8EC17926AB1135D7234F540A4486951E73967FC60C2D6D86B6230FF081ED34FEC3251FCDE5C4D'
 - (1013,48) INTEGER : '0A555CA2359A949C0F68C56BF7B72C1AD77108825B8053783A32F00BF685A2785EEECB5A1673A6ED6577A1B59560C4A4'

File Name: C:\zriha\data\CSCA_certificates\Switzerland.crt Size: 1063 (bytes)

ASN.1 – ECDSA signature OLD



ECDSA with SHA1	1.2.840.10045.1 (ANSI X9.62)
ECDSA with SHA1	1.2.840.10045.4.1 (ANSI X9.62)
ECDSA with SHA224	1.2.840.10045.4.3.1 (ANSI X9.62)
ECDSA with SHA256	1.2.840.10045.4.3.2 (ANSI X9.62)
ECDSA with SHA384	1.2.840.10045.4.3.3 (ANSI X9.62)
ECDSA with SHA512	1.2.840.10045.4.3.4 (ANSI X9.62)
ECDSA with SHA1	0.4.0.127.0.7.4.1.1 (BSI)
ECDSA with SHA224	0.4.0.127.0.7.4.1.2 (BSI)
ECDSA with SHA256	0.4.0.127.0.7.4.1.3 (BSI)
ECDSA with SHA384	0.4.0.127.0.7.4.1.4 (BSI)
ECDSA with SHA512	0.4.0.127.0.7.4.1.5 (BSI)



ASN.1 - certificates

```
Certificate ::= SEQUENCE {  
    tbsCertificate      TBSCertificate,  
    signatureAlgorithm  AlgorithmIdentifier,  
    signatureValue      BIT STRING }
```

```
TBSCertificate ::= SEQUENCE {  
    version             [0] EXPLICIT Version DEFAULT v1,  
    serialNumber        CertificateSerialNumber,  
    signature           AlgorithmIdentifier,  
    issuer              Name,  
    validity            Validity,  
    subject             Name,  
    subjectPublicKeyInfo SubjectPublicKeyInfo,  
    issuerUniqueID     [1] IMPLICIT UniqueIdentifier OPTIONAL,  
                      -- If present, version MUST be v2 or v3  
    subjectUniqueID    [2] IMPLICIT UniqueIdentifier OPTIONAL,  
                      -- If present, version MUST be v2 or v3  
    extensions         [3] EXPLICIT Extensions OPTIONAL  
                      -- If present, version MUST be v3  
}
```

```
Version ::= INTEGER { v1(0), v2(1), v3(2) }  
CertificateSerialNumber ::= INTEGER
```

Source:
RFC 5280



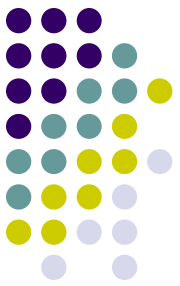
ASN.1 – certificates - pubkey

```
SubjectPublicKeyInfo ::= SEQUENCE {  
    algorithm      AlgorithmIdentifier,  
    subjectPublicKey BIT STRING }
```

Source:
RFC 5280

```
AlgorithmIdentifier ::= SEQUENCE {  
    algorithm      OBJECT IDENTIFIER,  
    parameters    ANY DEFINED BY algorithm OPTIONAL }
```

```
(293,418) SEQUENCE  
├── (297,13) SEQUENCE  
│   ├── (299,9) OBJECT IDENTIFIER : rsaEncryption : '1.2.840.113549.1.1.1'  
│   └── (310,0) NULL  
└── (312,399) BIT STRING UnusedBits: 0  
    ├── (317,394) SEQUENCE  
    │   ├── (321,385) INTEGER : '00A4A6BEDFA5969EE5647114F3E610CAB822C7B21098E6156CE073CCA6DA511E8F9AB6A1BD1DA64ED6B05'  
    │   └── (710,3) INTEGER : '65537'
```



ASN.1 – certificates - times

```
Validity ::= SEQUENCE {  
    notBefore      Time,  
    notAfter       Time }
```

Source:
RFC 5280

```
Time ::= CHOICE {  
    utcTime          UTCTime,  
    generalTime      GeneralizedTime }
```

- Until 2049: UTCTime
 - YYMMDDHHMMSSZ
- From 2050: GeneralizedTime
 - YYYYMMDDHHMMSSZ

 [CSCA_CZE.crt](#)

```
┌─── (172,30) SEQUENCE  
│   ┌─── (174,13) UTC TIME : '090113000000Z'  
│   └─── (189,13) UTC TIME : '240413000000Z'
```



ASN.1 – certificates - names

```
Name ::= CHOICE { -- only one possibility for now --  
    rdnSequence  RDNSequence }
```

```
RDNSequence ::= SEQUENCE OF RelativeDistinguishedName
```

```
RelativeDistinguishedName ::=  
    SET SIZE (1..MAX) OF AttributeTypeAndValue
```

```
AttributeTypeAndValue ::= SEQUENCE {  
    type      AttributeType,  
    value     AttributeValue }
```

```
AttributeType ::= OBJECT IDENTIFIER
```

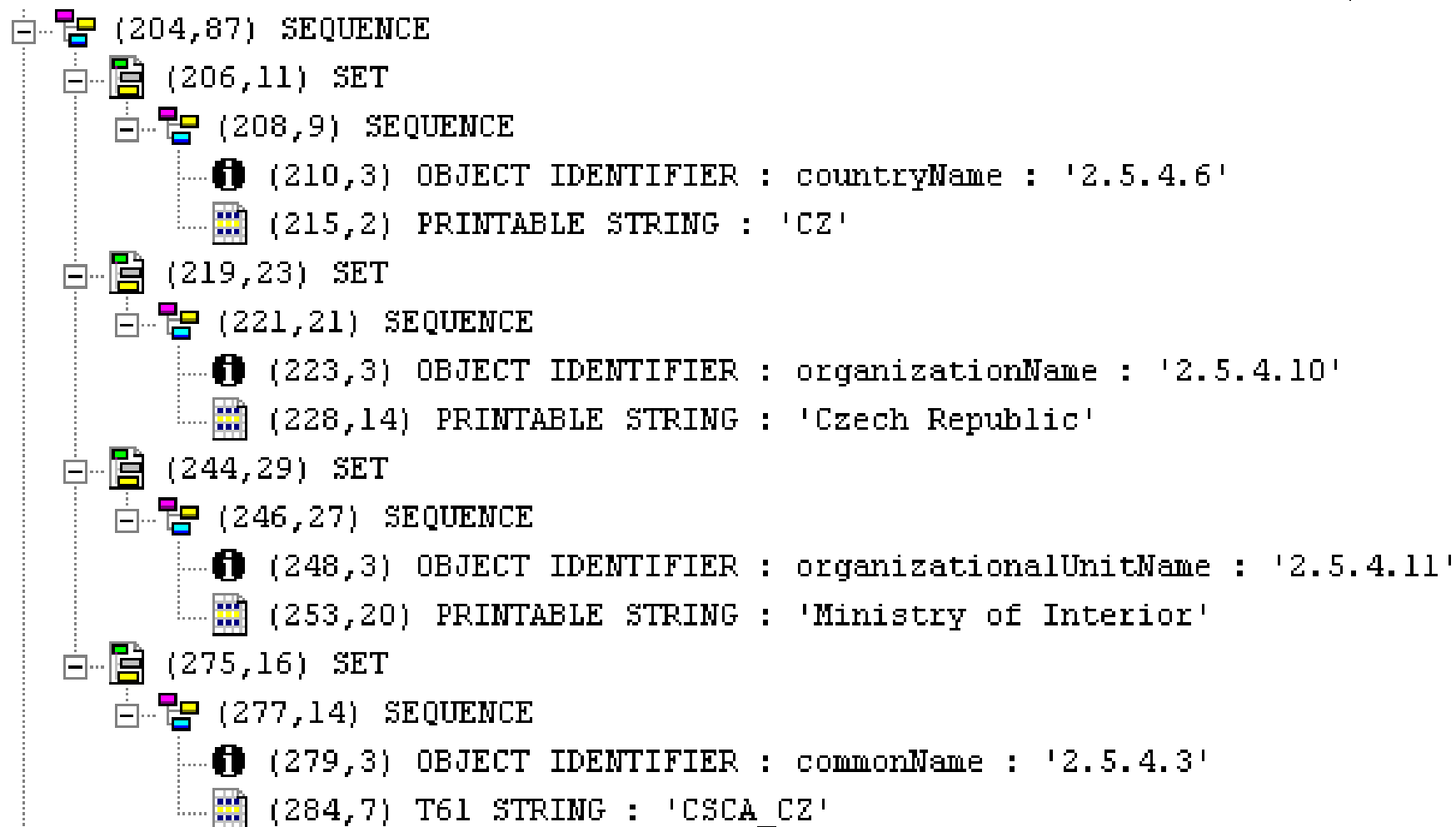
```
AttributeValue ::= ANY -- DEFINED BY AttributeType
```

```
DirectoryString ::= CHOICE {  
    teletexString      TeletexString (SIZE (1..MAX)),  
    printableString   PrintableString (SIZE (1..MAX)),  
    universalString    UniversalString (SIZE (1..MAX)),  
    utf8String         UTF8String (SIZE (1..MAX)),  
    bmpString          BMPString (SIZE (1..MAX)) }
```

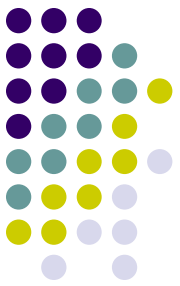
Source:
RFC 5280



ASN.1 – certificate - names



ASN.1 – certificate - names



```
commonName ATTRIBUTE ::= {
    SUBTYPE OF      name
    WITH SYNTAX     DirectoryString {ub-common-name}
    ID              id-at-commonName }
```

```
DirectoryString { INTEGER : maxSize } ::= CHOICE {
    teletexString TeletexString (SIZE (1..maxSize)),
    printableString PrintableString (SIZE (1..maxSize)),
    bmpString BMPString (SIZE (1..maxSize)),
    universalString UniversalString (SIZE (1..maxSize)),
    uTF8String UTF8String (SIZE (1..maxSize)) }
```

```
countryName ATTRIBUTE ::= {
    SUBTYPE OF      name
    WITH SYNTAX     CountryName
    SINGLE VALUE   TRUE
    ID              id-at-countryName }
```

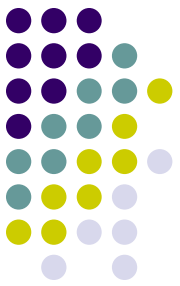
```
CountryName ::= PrintableString (SIZE(2))
```

```
-- id-at-objectClass
-- id-at-aliasedEntryName
-- id-at-encryptedAliasedEntryName
id-at-knowledgeInformation
id-at-commonName
-- id-at-encryptedCommonName
id-at-surname
-- id-at-encryptedSurname
id-at-serialNumber
-- id-at-encryptedSerialNumber
id-at-countryName
```

```
OBJECT IDENTIFIER ::= {id-at 0}
OBJECT IDENTIFIER ::= {id-at 1}
OBJECT IDENTIFIER ::= {id-at 1 2}
OBJECT IDENTIFIER ::= {id-at 2}
OBJECT IDENTIFIER ::= {id-at 3}
OBJECT IDENTIFIER ::= {id-at 3 2}
OBJECT IDENTIFIER ::= {id-at 4}
OBJECT IDENTIFIER ::= {id-at 4 2}
OBJECT IDENTIFIER ::= {id-at 5}
OBJECT IDENTIFIER ::= {id-at 5 2}
OBJECT IDENTIFIER ::= {id-at 6}
```

Source:
ITU-T X.520

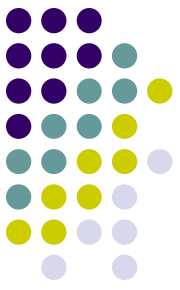
ASN.1 – certificate - names



id-at-localityName
-- *id-at-encryptedLocalityName*
id-at-collectiveLocalityName
-- *id-at-encryptedCollectiveLocalityName*
id-at-stateOrProvinceName
-- *id-at-encryptedStateOrProvinceName*
id-at-collectiveStateOrProvinceName
-- *id-at-encryptedCollectiveStateOrProvinceName*
id-at-streetAddress
-- *id-at-encryptedStreetAddress*
id-at-collectiveStreetAddress
-- *id-at-encryptedCollectiveStreetAddress*
id-at-organizationName
-- *id-at-encryptedOrganizationName*
id-at-collectiveOrganizationName
-- *id-at-encryptedCollectiveOrganizationName*
id-at-organizationalUnitName
-- *id-at-encryptedOrganizationalUnitName*
id-at-collectiveOrganizationalUnitName
-- *id-at-encryptedCollectiveOrganizationalUnitName*
id-at-title
-- *id-at-encryptedTitle*
id-at-description
-- *id-at-encryptedDescription*
id-at-searchGuide
-- *id-at-encryptedSearchGuide*
id-at-businessCategory
-- *id-at-encryptedBusinessCategory*
id-at-postalAddress
-- *id-at-encryptedPostalAddress*
id-at-collectivePostalAddress
-- *id-at-encryptedCollectivePostalAddress*
id-at-postalCode
-- *id-at-encryptedPostalCode*
id-at-collectivePostalCode
-- *id-at-encryptedCollectivePostalCode*

OBJECT IDENTIFIER ::= {id-at 7}
OBJECT IDENTIFIER ::= {id-at 7 2}
OBJECT IDENTIFIER ::= {id-at 7 1}
OBJECT IDENTIFIER ::= {id-at 7 1 2}
OBJECT IDENTIFIER ::= {id-at 8}
OBJECT IDENTIFIER ::= {id-at 8 2}
OBJECT IDENTIFIER ::= {id-at 8 1}
OBJECT IDENTIFIER ::= {id-at 8 1 2}
OBJECT IDENTIFIER ::= {id-at 9}
OBJECT IDENTIFIER ::= {id-at 9 2}
OBJECT IDENTIFIER ::= {id-at 9 1}
OBJECT IDENTIFIER ::= {id-at 9 1 2}
OBJECT IDENTIFIER ::= {id-at 10}
OBJECT IDENTIFIER ::= {id-at 10 2}
OBJECT IDENTIFIER ::= {id-at 10 1}
OBJECT IDENTIFIER ::= {id-at 10 1 2}
OBJECT IDENTIFIER ::= {id-at 11}
OBJECT IDENTIFIER ::= {id-at 11 2}
OBJECT IDENTIFIER ::= {id-at 11 1}
OBJECT IDENTIFIER ::= {id-at 11 1 2}
OBJECT IDENTIFIER ::= {id-at 12}
OBJECT IDENTIFIER ::= {id-at 12 2}
OBJECT IDENTIFIER ::= {id-at 13}
OBJECT IDENTIFIER ::= {id-at 13 2}
OBJECT IDENTIFIER ::= {id-at 14}
OBJECT IDENTIFIER ::= {id-at 14 2}
OBJECT IDENTIFIER ::= {id-at 15}
OBJECT IDENTIFIER ::= {id-at 15 2}
OBJECT IDENTIFIER ::= {id-at 16}
OBJECT IDENTIFIER ::= {id-at 16 2}
OBJECT IDENTIFIER ::= {id-at 16 1}
OBJECT IDENTIFIER ::= {id-at 16 1 2}
OBJECT IDENTIFIER ::= {id-at 17}
OBJECT IDENTIFIER ::= {id-at 17 2}
OBJECT IDENTIFIER ::= {id-at 17 1}
OBJECT IDENTIFIER ::= {id-at 17 1 2}

Source:
ITU-T X.520



Certificate profiles

- For particular areas/purposes there exist certificate profiles which prescribe what kind of attributes will be used in Names
- E.g. for electronic passports ICAO Doc. 9303 states:

The following Attributes SHOULD be used:

- country (country codes SHALL follow the format of two letter country codes, specified in [R16], *ISO/IEC 3166, Codes for the representation of names of countries and their subdivisions — 1997.*).
- organization;
- organizational-unit;
- common name.

Additionally some countries MAY use:

- serial number.

Source:
ICAO Doc. 9303



ASN.1 – certificates – v3

```
UniqueIdentifier ::= BIT STRING
```

```
Extensions ::= SEQUENCE SIZE (1..MAX) OF Extension
```

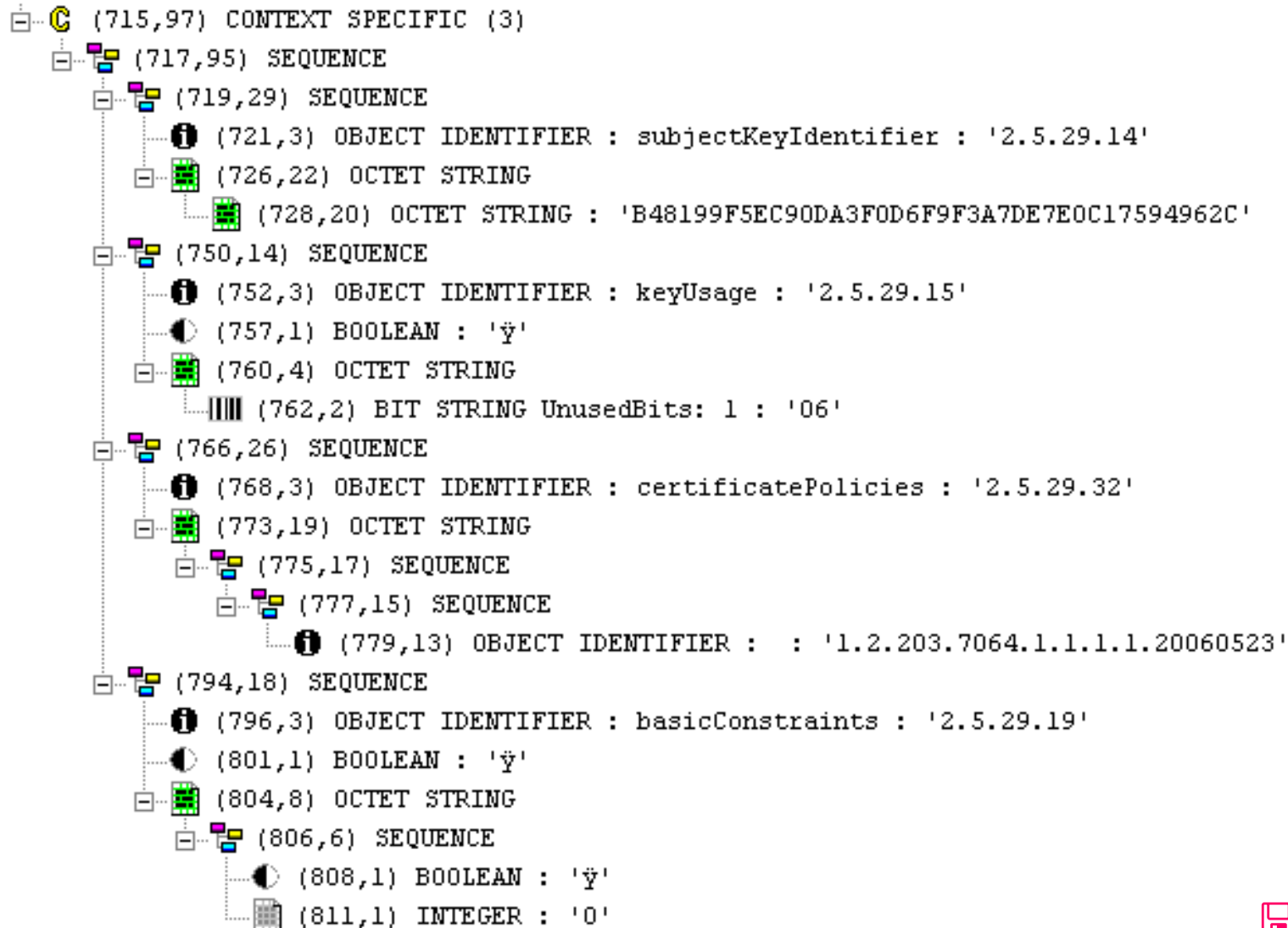
```
Extension ::= SEQUENCE {  
    extnID          OBJECT IDENTIFIER,  
    critical        BOOLEAN DEFAULT FALSE,  
    extnValue       OCTET STRING  
    -- contains the DER encoding of an ASN.1 value  
    -- corresponding to the extension type identified  
    -- by extnID  
}
```

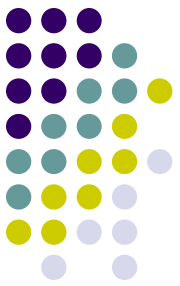
Source:
RFC 5280

- Critical x non-critical extensions



ASN.1 – certs – extensions





X509v3 cert extensions

- Authority Key Identifier
 - Identification of the issuing CA
 - Non critical

```
id-ce-authorityKeyIdentifier OBJECT IDENTIFIER ::= { id-ce 35 }
```

```
AuthorityKeyIdentifier ::= SEQUENCE {  
    keyIdentifier          [0] KeyIdentifier          OPTIONAL,  
    authorityCertIssuer    [1] GeneralNames          OPTIONAL,  
    authorityCertSerialNumber [2] CertificateSerialNumber OPTIONAL }
```

```
KeyIdentifier ::= OCTET STRING
```

- Similarly “Subject Key Identifier”

Source:
RFC 5280



X509v3 cert extensions

- Key Usage
 - Restrictions of the use of the key

```
id-ce-keyUsage OBJECT IDENTIFIER ::= { id-ce 15 }
```

```
KeyUsage ::= BIT STRING {  
    digitalSignature          (0),  
    nonRepudiation           (1), -- recent editions of X.509 have  
                                -- renamed this bit to contentCommitment  
    keyEncipherment          (2),  
    dataEncipherment         (3),  
    keyAgreement             (4),  
    keyCertSign              (5),  
    cRLSign                  (6),  
    encipherOnly             (7),  
    decipherOnly             (8) }
```

Source:
RFC 5280



X509v3 cert extensions

- Extended Key Usage
 - Purposes of the certified key

```
id-ce-extKeyUsage OBJECT IDENTIFIER ::= { id-ce 37 }
ExtKeyUsageSyntax ::= SEQUENCE SIZE (1..MAX) OF KeyPurposeId
KeyPurposeId ::= OBJECT IDENTIFIER
anyExtendedKeyUsage OBJECT IDENTIFIER ::= { id-ce-extKeyUsage 0 }
```

```
id-kp OBJECT IDENTIFIER ::= { id-pkix 3 }
id-kp-serverAuth OBJECT IDENTIFIER ::= { id-kp 1 }
id-kp-clientAuth OBJECT IDENTIFIER ::= { id-kp 2 }
id-kp-codeSigning OBJECT IDENTIFIER ::= { id-kp 3 }
id-kp-emailProtection OBJECT IDENTIFIER ::= { id-kp 4 }
id-kp-timeStamping OBJECT IDENTIFIER ::= { id-kp 8 }
id-kp-OCSPSigning OBJECT IDENTIFIER ::= { id-kp 9 }
```

X509v3 cert extensions



```
id-ce-certificatePolicies OBJECT IDENTIFIER ::= { id-ce 32 }
anyPolicy OBJECT IDENTIFIER ::= { id-ce-certificatePolicies 0 }

certificatePolicies ::= SEQUENCE SIZE (1..MAX) OF PolicyInformation

PolicyInformation ::= SEQUENCE {
    policyIdentifier    CertPolicyId,
    policyQualifiers   SEQUENCE SIZE (1..MAX) OF
                        PolicyQualifierInfo OPTIONAL }

CertPolicyId ::= OBJECT IDENTIFIER

PolicyQualifierInfo ::= SEQUENCE {
    policyQualifierId  PolicyQualifierId,
    qualifier          ANY DEFINED BY policyQualifierId }

-- policyQualifierIds for Internet policy qualifiers

id-qt          OBJECT IDENTIFIER ::= { id-pkix 2 }
id-qt-cps      OBJECT IDENTIFIER ::= { id-qt 1 }
id-qt-unotice  OBJECT IDENTIFIER ::= { id-qt 2 }

PolicyQualifierId ::= OBJECT IDENTIFIER ( id-qt-cps | id-qt-unotice )

Qualifier ::= CHOICE {
    cpsSuri          CPSuri,
    userNotice       UserNotice }

CPSuri ::= IA5String

UserNotice ::= SEQUENCE {
    noticeRef        NoticeReference OPTIONAL,
    explicitText     DisplayText OPTIONAL }

NoticeReference ::= SEQUENCE {
    organization     DisplayText,
    noticeNumbers    SEQUENCE OF INTEGER }

DisplayText ::= CHOICE {
    ia5String        IA5String          (SIZE (1..200)),
    visibleString    VisibleString      (SIZE (1..200)),
    bmpString        BMPString          (SIZE (1..200)),
    utf8String       UTF8String         (SIZE (1..200)) }
```

- Certificate Policies
 - Policy relevant for the issue and use of the certificate
 - Preferably only an OID

Source:
RFC 5280



X509v3 cert extensions

- Subject Alternative Name
- Issuer Alternative Name
- “Internet style identities”
 - Email
 - DNS name
 - IP address
 - URL
- Must be verified by CA



X509v3 cert extensions

- Basic Constraints
- Is Subject a CA?
- Max. length/depth of the certificate chain/path
 - A pathLenConstraint of zero indicates that no non-self-issued intermediate CA certificates may follow in a valid certification path.

```
id-ce-basicConstraints OBJECT IDENTIFIER ::= { id-ce 19 }
```

```
BasicConstraints ::= SEQUENCE {  
    cA                BOOLEAN DEFAULT FALSE,  
    pathLenConstraint INTEGER (0..MAX) OPTIONAL }
```



X509v3 cert extensions

- Name Constraints
- Only for CA certificates
- “indicates a name space within which all subject names in subsequent certificates in a certification path MUST be located”

```
id-ce-nameConstraints OBJECT IDENTIFIER ::= { id-ce 30 }

NameConstraints ::= SEQUENCE {
    permittedSubtrees    [0]    GeneralSubtrees OPTIONAL,
    excludedSubtrees     [1]    GeneralSubtrees OPTIONAL }

GeneralSubtrees ::= SEQUENCE SIZE (1..MAX) OF GeneralSubtree

GeneralSubtree ::= SEQUENCE {
    base                GeneralName,
    minimum             [0]    BaseDistance DEFAULT 0,
    maximum             [1]    BaseDistance OPTIONAL }

BaseDistance ::= INTEGER (0..MAX)
```

Source:
RFC 5280



X509v3 cert extensions

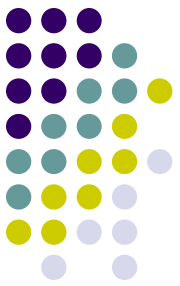
- Policy Constraints
- Must be critical
- For CA certificates
- Constraints path validation
 - Prohibit policy mapping (or)
 - Require acceptable policy OID in each certificate

```
id-ce-policyConstraints OBJECT IDENTIFIER ::= { id-ce 36 }
```

```
PolicyConstraints ::= SEQUENCE {  
    requireExplicitPolicy          [0] SkipCerts OPTIONAL,  
    inhibitPolicyMapping          [1] SkipCerts OPTIONAL }
```

```
SkipCerts ::= INTEGER (0..MAX)
```

Source:
RFC 5280



X509v3 cert extensions

- CRL Distribution Points
- How to obtain CRL

```
id-ce-cRLDistributionPoints OBJECT IDENTIFIER ::= { id-ce 31 }
```

```
CRLDistributionPoints ::= SEQUENCE SIZE (1..MAX) OF DistributionPoint
```

```
DistributionPoint ::= SEQUENCE {  
    distributionPoint [0] DistributionPointName OPTIONAL,  
    reasons [1] ReasonFlags OPTIONAL,  
    cRLIssuer [2] GeneralNames OPTIONAL }
```

```
DistributionPointName ::= CHOICE {  
    fullName [0] GeneralNames,  
    nameRelativeToCRLIssuer [1] RelativeDistinguishedName }
```

```
ReasonFlags ::= BIT STRING {  
    unused (0),  
    keyCompromise (1),  
    cACompromise (2),  
    affiliationChanged (3),  
    superseded (4),  
    cessationOfOperation (5),  
    certificateHold (6),  
    privilegeWithdrawn (7),  
    aACompromise (8) }
```

Source:
RFC 5280



ASN.1 – certificate request

```
CertificationRequest ::= SEQUENCE {  
  certificationRequestInfo CertificationRequestInfo,  
  signatureAlgorithm AlgorithmIdentifier,  
  signature BIT STRING  
}
```

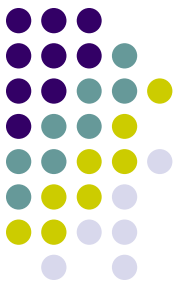
```
CertificationRequestInfo ::= SEQUENCE {  
  version INTEGER { v1(0) },  
  subject Name,  
  subjectPKInfo SubjectPublicKeyInfo,  
  attributes [0] Attributes  
}
```

```
Attributes ::= SET OF Attribute
```

```
Attribute ::= SEQUENCE {  
  type ATTRIBUTE.&id({IOSet}),  
  values SET SIZE(1..MAX) OF ATTRIBUTE.&Type({IOSet}@type)  
}
```

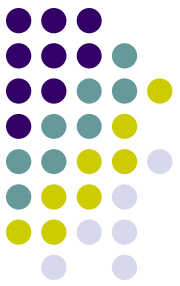
Source:
RFC 5280

ASN.1 - CRL



```
CertificateList ::= SEQUENCE {  
    tbsCertList      TBSCertList,  
    signatureAlgorithm AlgorithmIdentifier,  
    signatureValue   BIT STRING }
```

```
TBSCertList ::= SEQUENCE {  
    version          Version OPTIONAL,  
                    -- if present, MUST be v2  
    signature        AlgorithmIdentifier,  
    issuer           Name,  
    thisUpdate      Time,  
    nextUpdate      Time OPTIONAL,  
    revokedCertificates SEQUENCE OF SEQUENCE {  
        userCertificate      CertificateSerialNumber,  
        revocationDate      Time,  
        crlEntryExtensions  Extensions OPTIONAL  
                    -- if present, version MUST be v2  
    } OPTIONAL,  
    crlExtensions [0] EXPLICIT Extensions OPTIONAL  
                    -- if present, version MUST be v2  
}
```



ASN.1 – PKCS#7 / CMS

```
ContentInfo ::= SEQUENCE {  
    contentType ContentType,  
    content [0] EXPLICIT ANY DEFINED BY contentType }
```

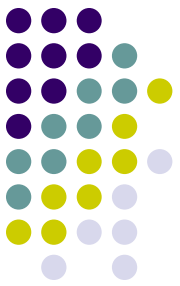
```
ContentType ::= OBJECT IDENTIFIER
```

```
SignedData ::= SEQUENCE {  
    version CMSVersion,  
    digestAlgorithms DigestAlgorithmIdentifiers,  
    encapContentInfo EncapsulatedContentInfo,  
    certificates [0] IMPLICIT CertificateSet OPTIONAL,  
    crls [1] IMPLICIT RevocationInfoChoices OPTIONAL,  
    signerInfos SignerInfos }
```

```
DigestAlgorithmIdentifiers ::= SET OF DigestAlgorithmIdentifier
```

```
EncapsulatedContentInfo ::= SEQUENCE {  
    eContentType ContentType,  
    eContent [0] EXPLICIT OCTET STRING OPTIONAL }
```

```
SignerInfos ::= SET OF SignerInfo
```

ASN.1 - PKCS#7 / CMS

```
SignerInfo ::= SEQUENCE {  
    version CMSVersion,  
    sid SignerIdentifier,  
    digestAlgorithm DigestAlgorithmIdentifier,  
    signedAttrs [0] IMPLICIT SignedAttributes OPTIONAL,  
    signatureAlgorithm SignatureAlgorithmIdentifier,  
    signature SignatureValue,  
    unsignedAttrs [1] IMPLICIT UnsignedAttributes OPTIONAL }
```

```
SignerIdentifier ::= CHOICE {  
    issuerAndSerialNumber IssuerAndSerialNumber,  
    subjectKeyIdentifier [0] SubjectKeyIdentifier }
```

```
SignedAttributes ::= SET SIZE (1..MAX) OF Attribute
```

```
UnsignedAttributes ::= SET SIZE (1..MAX) OF Attribute
```

```
Attribute ::= SEQUENCE {  
    attrType OBJECT IDENTIFIER,  
    attrValues SET OF AttributeValue }
```

```
AttributeValue ::= ANY
```

```
SignatureValue ::= OCTET STRING
```




ASN.1 – PKCS#8

```
-- Private-key information syntax
```

```
PrivateKeyInfo ::= SEQUENCE {  
    version Version,  
    privateKeyAlgorithm AlgorithmIdentifier,  
    privateKey PrivateKey,  
    attributes [0] Attributes OPTIONAL }
```

```
Version ::= INTEGER {v1(0)} (v1,...)
```

```
PrivateKey ::= OCTET STRING
```

```
Attributes ::= SET OF Attribute
```

```
-- Encrypted private-key information syntax
```

```
EncryptedPrivateKeyInfo ::= SEQUENCE {  
    encryptionAlgorithm AlgorithmIdentifier,  
    encryptedData EncryptedData  
}
```

```
EncryptedData ::= OCTET STRING
```