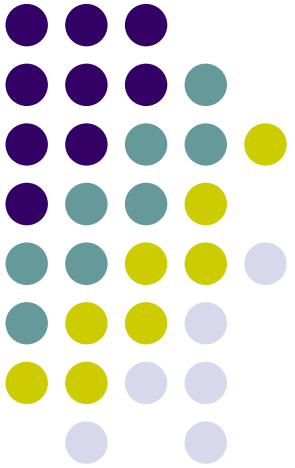
ASN.1:
Introduction

Zdeněk Říha



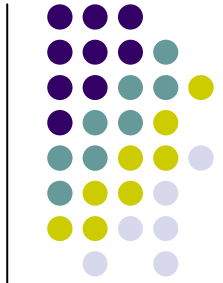
ASN.1

- 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

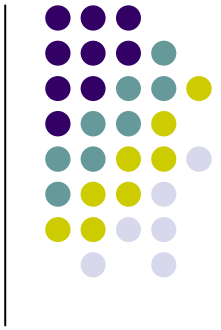


Type	Tag number (decimal)	Tag number (hexadecimal)
INTEGER	2	02
BIT STRING	3	03
OCTET STRING	4	04
NULL	5	05
OBJECT IDENTIFIER	6	06
SEQUENCE and SEQUENCE OF	16	10
SET and SET OF	17	11
PrintableString	19	13
IA5String	22	16
UTCTime	23	17

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)
- PringtableString, 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

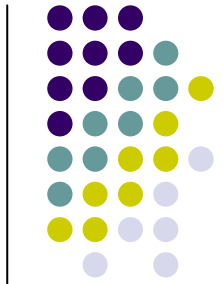
ASN.1 – structured types

- 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)

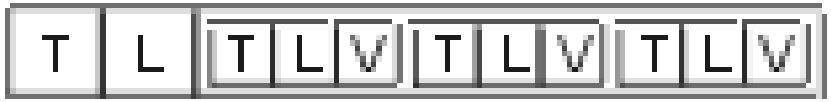
BER encoding

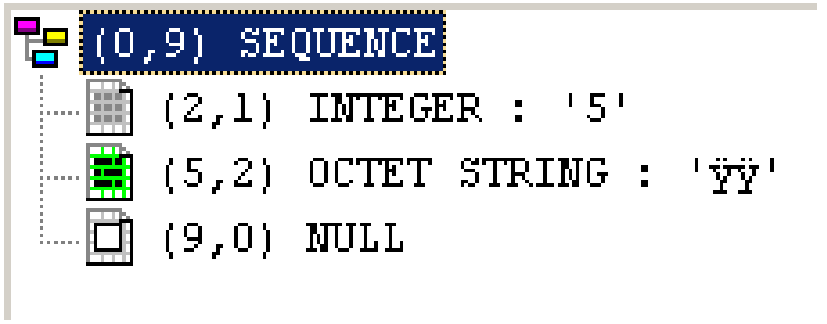


- TLV – Tag Length Value
  - All the data is encoded using a simple TLV format
  - Tag – what kind of data it is
  - Length – the length of the data
  - Value – the data itself
- Example
  - 02 01 05 [hexadecimal values]
    - Tag – Integer
    - Length of data – 1 byte
    - Data: (positive integer)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

Nested data

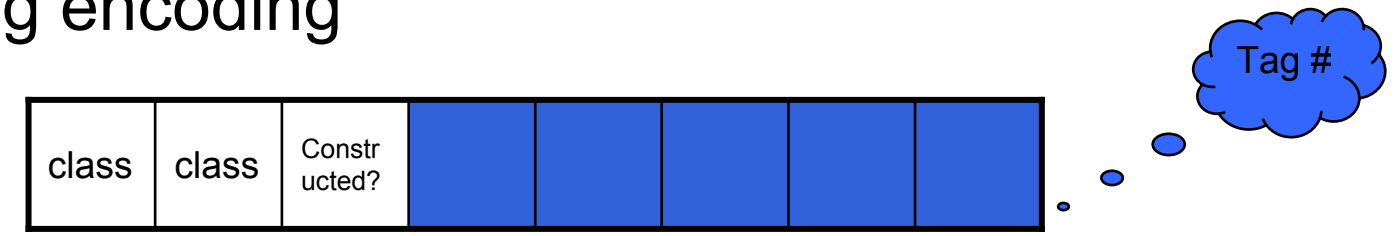
- SEQUENCE is similar to struct/record
- 30 09 02 01 05 04 02 FF FF 05 00
  - 30 09 – sequence of length 9 bytes
  - 02 01 05 – integer 5
  - 04 02 FF FF – octet string FF FF
  - 05 00 – NULL (no data)





BER tags

- Tag encoding

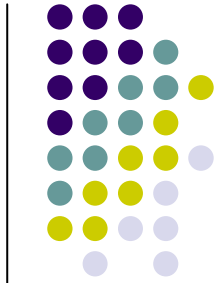


- Class

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

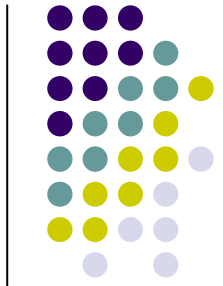
- Tag number
  - Bits 1-5
  - If all bits are 1 then the tag continues in the following byte(s)

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

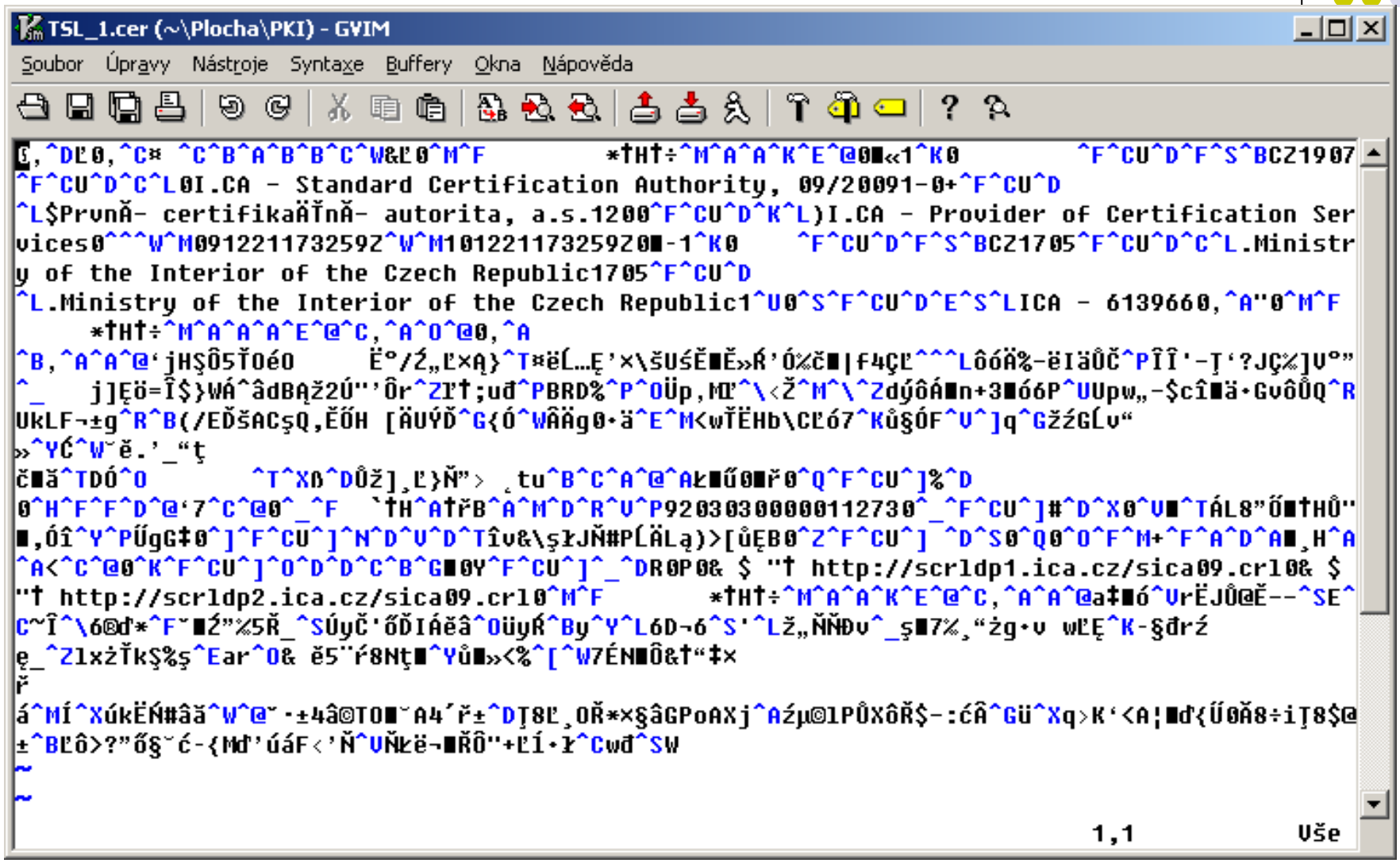
BER value



- 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

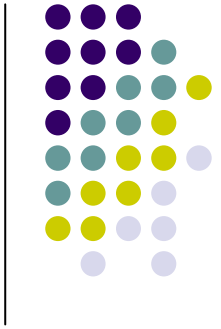
YYMMDDhhmmZ
YYMMDDhhmm+hh'mm'
YYMMDDhhmm-hh'mm'
YYMMDDhhmmssZ
YYMMDDhhmmss+hh'mm'
YYMMDDhhmmss-hh'mm'

First look at the binary DER file



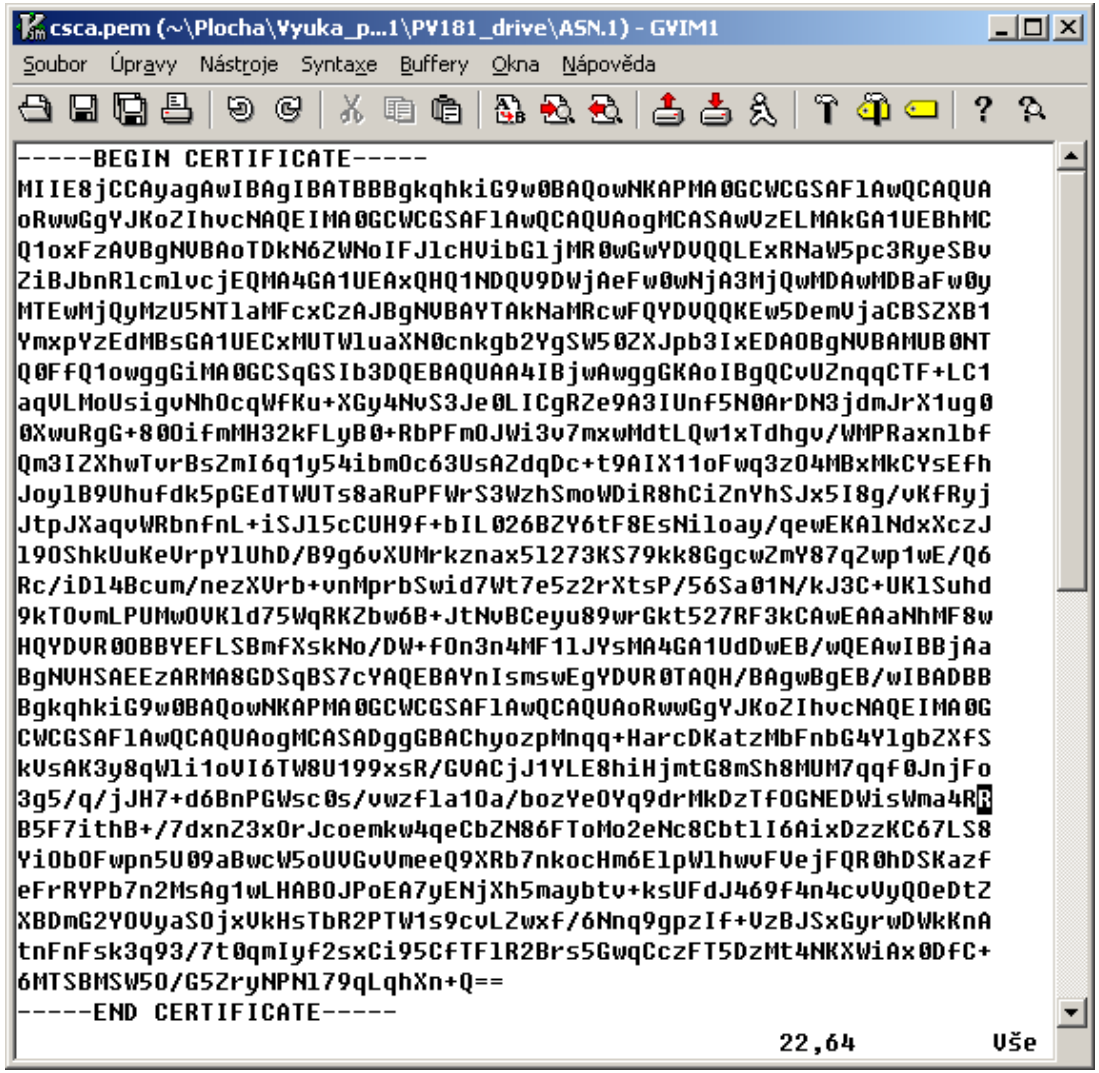
CSCA_CZE.crt

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

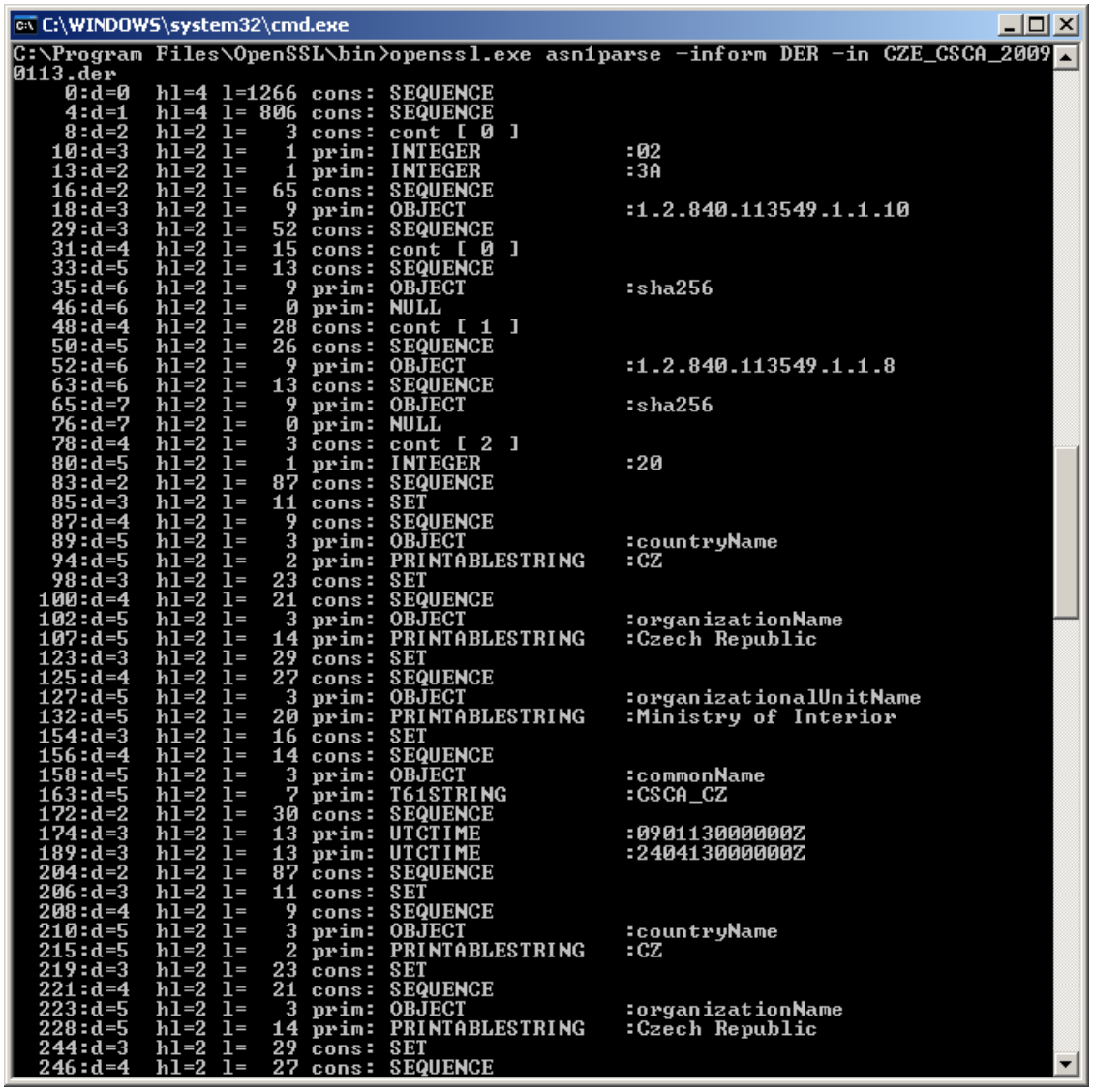


CSCA_CZE.pem

ASN.1 viewers

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

OpenSSL asn1parse



CSCA_CZE.crt

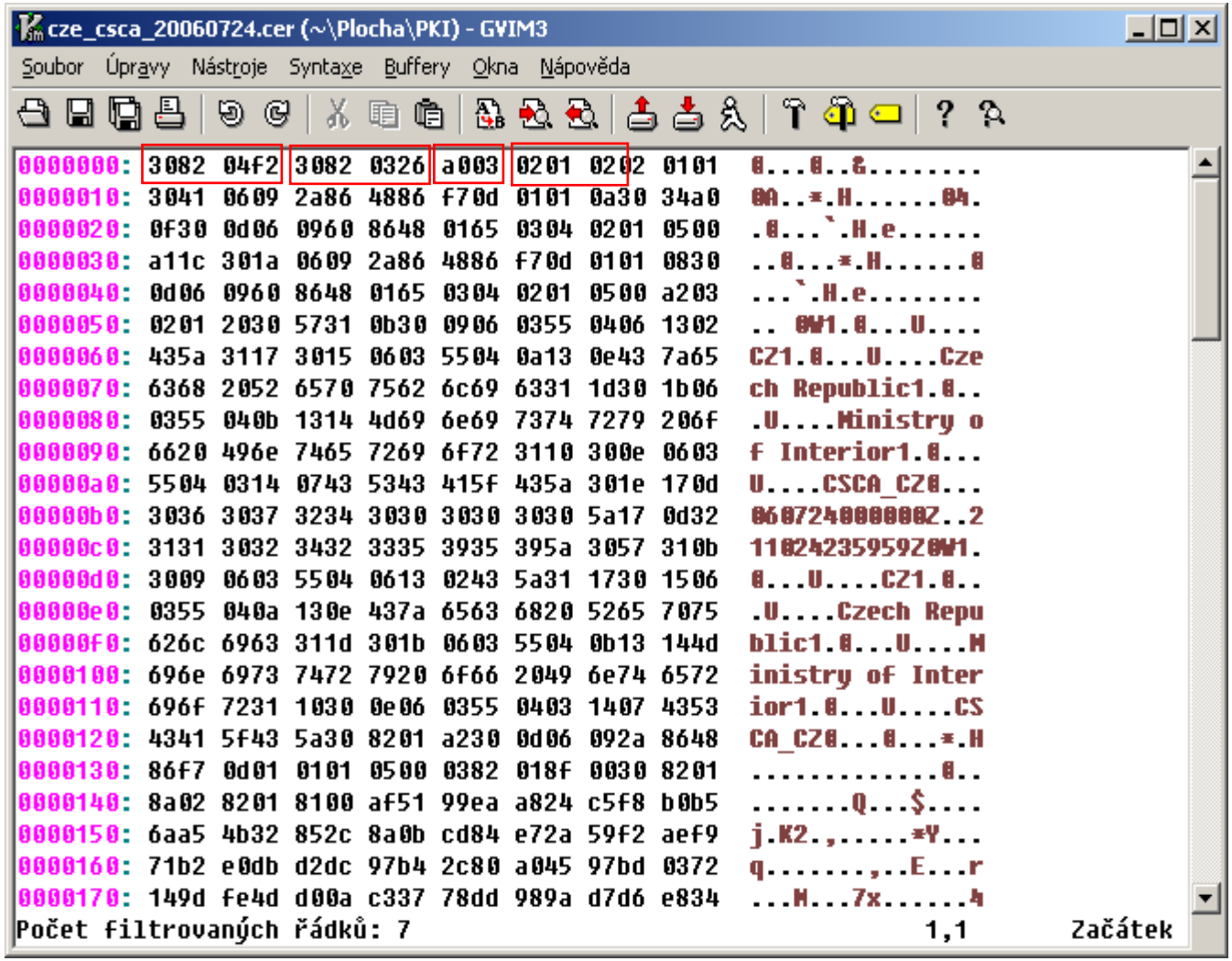
unber

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 O="0" T="[UNIVERSAL 16]" TL="4" V="952" A="SEQUENCE">
    <C O="4" T="[UNIVERSAL 16]" TL="4" V="804" A="SEQUENCE">
        <C O="8" T="[0]" TL="2" V="3">
            <P O="10" T="[UNIVERSAL 2]" TL="2" V="1" A="INTEGER" F>2</P>
        </C O="13" T="[0]" L="5">
        <P O="13" T="[UNIVERSAL 2]" TL="2" V="1" A="INTEGER" F>53</P>
        <C O="16" T="[UNIVERSAL 16]" TL="2" V="10" A="SEQUENCE">
            <P O="18" T="[UNIVERSAL 6]" TL="2" V="6" A="OBJECT IDENTIFIER" F>1.3.36.3.3.1.2</P>
            <P O="26" T="[UNIVERSAL 5]" TL="2" V="0" A="NULL"></P>
        </C O="28" T="[UNIVERSAL 16]" A="SEQUENCE" L="12">
        <C O="28" T="[UNIVERSAL 16]" TL="2" V="63" A="SEQUENCE">
            <C O="30" T="[UNIVERSAL 17]" TL="2" V="11" A="SET">
                <C O="32" T="[UNIVERSAL 16]" TL="2" V="9" A="SEQUENCE">
                    <P O="34" T="[UNIVERSAL 6]" TL="2" V="3" A="OBJECT IDENTIFIER" F>2.5.4.6</P>
                    <P O="39" T="[UNIVERSAL 19]" TL="2" V="2" A="PrintableString">DE</P>
                </C O="43" T="[UNIVERSAL 16]" A="SEQUENCE" L="11">
            </C O="43" T="[UNIVERSAL 17]" A="SET" L="13">
            <C O="43" T="[UNIVERSAL 17]" TL="2" V="26" A="SET">
                <C O="45" T="[UNIVERSAL 16]" TL="2" V="24" A="SEQUENCE">
                    <P O="47" T="[UNIVERSAL 6]" TL="2" V="3" A="OBJECT IDENTIFIER" F>2.5.4.10</P>
                    <P O="52" T="[UNIVERSAL 12]" TL="2" V="17" A="UTF8String">Bundesnetzagentur</P>
                </C O="71" T="[UNIVERSAL 16]" A="SEQUENCE" L="26">
            </C O="71" T="[UNIVERSAL 17]" A="SET" L="28">
            <C O="71" T="[UNIVERSAL 17]" TL="2" V="20" A="SET">
                <C O="73" T="[UNIVERSAL 16]" TL="2" V="18" A="SEQUENCE">
                    <P O="75" T="[UNIVERSAL 6]" TL="2" V="3" A="OBJECT IDENTIFIER" F>2.5.4.3</P>
                    <P O="80" T="[UNIVERSAL 12]" TL="2" V="11" A="UTF8String">10R-CA 1:PN</P>
                </C O="93" T="[UNIVERSAL 16]" A="SEQUENCE" L="20">
            </C O="93" T="[UNIVERSAL 17]" A="SET" L="22">
        </C O="93" T="[UNIVERSAL 16]" A="SEQUENCE" L="65">
        <C O="93" T="[UNIVERSAL 16]" TL="2" V="30" A="SEQUENCE">
            <P O="95" T="[UNIVERSAL 23]" TL="2" V="13" A="UTCTime">050804082709Z</P>
            <P O="110" T="[UNIVERSAL 23]" TL="2" V="13" A="UTCTime">071231082349Z</P>
        </C O="125" T="[UNIVERSAL 16]" A="SEQUENCE" L="32">
        <C O="125" T="[UNIVERSAL 16]" TL="2" V="65" A="SEQUENCE">
            <C O="127" T="[UNIVERSAL 17]" TL="2" V="11" A="SET">
                <C O="129" T="[UNIVERSAL 16]" TL="2" V="9" A="SEQUENCE">
                    <P O="131" T="[UNIVERSAL 6]" TL="2" V="3" A="OBJECT IDENTIFIER" F>2.5.4.6</P>
                    <P O="136" T="[UNIVERSAL 19]" TL="2" V="2" A="PrintableString">DE</P>
                </C O="140" T="[UNIVERSAL 16]" A="SEQUENCE" L="11">
            </C O="140" T="[UNIVERSAL 17]" A="SET" L="13">
            <C O="140" T="[UNIVERSAL 17]" TL="2" V="26" A="SET">
                <C O="142" T="[UNIVERSAL 16]" TL="2" V="24" A="SEQUENCE">
                    <P O="144" T="[UNIVERSAL 6]" TL="2" V="3" A="OBJECT IDENTIFIER" F>2.5.4.10</P>
                    <P O="149" T="[UNIVERSAL 12]" TL="2" V="17" A="UTF8String">Bundesnetzagentur</P>
                </C O="168" T="[UNIVERSAL 16]" A="SEQUENCE" L="26">
```

Manual viewing/processing

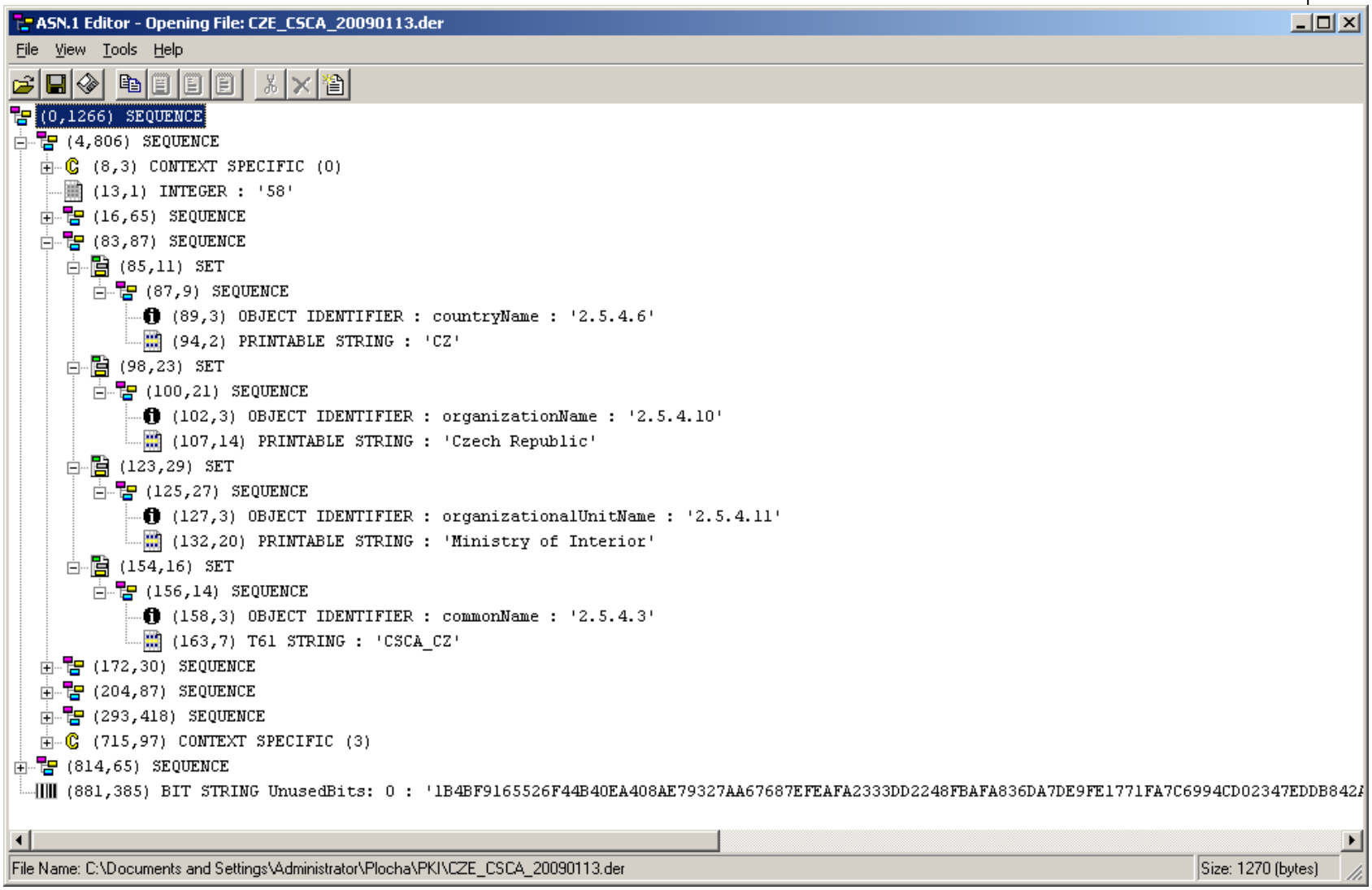


- 30 82 04 f2
  - SEQUENCE
  - length 1266B
- 30 82 03 26
  - SEQUENCE
  - length 806B
- A0 03
  - CONTEXT SPECIFIC 0
  - Length 3B
- 02 01 02
  - INTEGER: 2

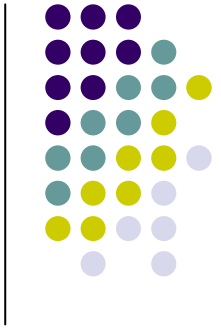
CSCA_CZE.crt

ASN.1 Editor

CSCA_CZE.crt



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
                                  }
```