

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

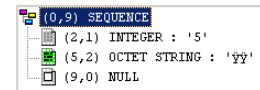
- XML – oriented formats
 - XER (XML Encoding Rules)
- Byte-oriented formats
 - BER (Basic Encoding Rules)
 - CER (Canonical Encoding Rules) – subset of BER
 - DER (Distinguished Encoding Rules) – subset of BER
 - Used for crypto files
- Bit-oriented formats
 - PER (Packed Encoding Rules)
- Verbose,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

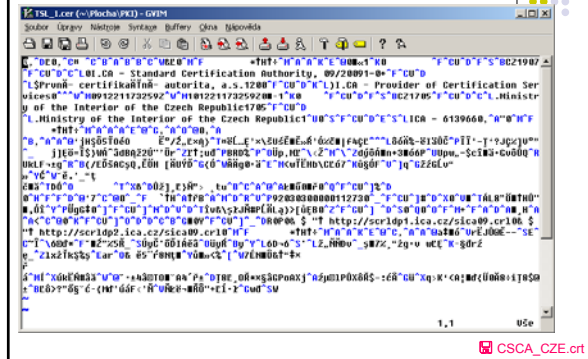
- length ≥ 0 && length ≤ 127
 - The length is coded directly
 - E.g. '05'
- Otherwise the bit 8 is set, bits 1-7 code the number of bytes that specify the length
 - E.g. 255 -> '81' 'FF'
 - E.g. 256 -> '82' '01' '00' or also '83' '00' '01' '00'
 - BER x DER
- '80' is "indefinite" length
 - Not allowed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



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

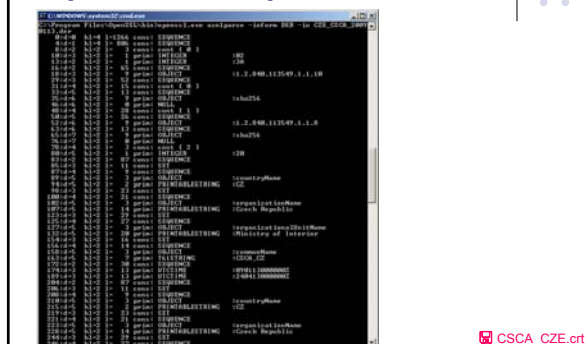


CSCA_CZE.pem

ASN.1 viewers

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

OpenSSL asn1parse



CSCA_CZE.crt

