Home Assignment 5: A-part

11 december 2017

1 Questions

1.1 A-2 In ASN.1, what is the difference between implicit and explicit tagging?

The difference is that in implicit tagging the context-specific tag *replaces* the universal tag while in explicit tagging you *add* an outer tag environment to the original tag.

For example: studentname ::= [1] IMPLICIT INTEGER says that for studentname with an implicit tag to replace the existing tag on INTEGER with [1]. studentname ::= [1] EXPLICIT INTEGER tells us that for studentname with an explicit tag, add [1] in front of the existing tag.

1.2 A-4 In ASN.1, describe the difference between SET and SEQUENCE and explain why one is usually the better choice.

The SEQUENCE type is an ordered list of component types i.e. the values must be given in the specified order. The SET type is almost the same but differs in that it is an unordered list of component types which means that the values appears in any order. Using SEQUENCE is frequently preferred over SET since it can make the decoding faster.

1.3 A-7 Consider the example on page 10 in the lecture notes where PER requires only one octet to represent (a,b,c) = (a,b,(c1,c2,c3)), but DER requires several octets. Give both the DER encoding and the PER encoding for the case where (a,b,c) = (TRUE, 30, (TRUE, FALSE, 50)).

DER encoding:
01 01 FF 02 01 1E 30 09 01 01 FF 01 01 00 02 01 32

PER encoding:
FF 01 1E FF 00 01 32

In PER encoding it is possible to omit the type and length from the TLV but if there are no constraints on INTEGER there must be some length encoding of the type.

1.4 A-8 For each of BER, CER, DER, PER, and XER, give the full name and summarize the main idea behind that particular encoding rule.

BER stands for Basic Encoding Rules, it uses a Tag-Length-Value (TLV) format for encoding all information. Moreover, BER sends a tag in order to indicate what kind of data follows then a length indicating the length of the data that follows and finally the value which is the actual data.

CER stands for Canonical Encoding Rules and DER stands for Distinguished Encoding Rules. Both are subsets of BER and canonical, which means they describe unique encoding for the data.

The meaning of PER is Packed Encoding Rules and differs from BER since it does not send the Tag and the length of the TLV if the Value has a fixed length. PER is able to do this since the order in which components of the message occur are known. Thus this process is making the PER messages compact and therefore it is known as the most compact of the encoding rules.

On the other hand, XER, is the least compact of the encoding rules. The syntax uses a XML format for its encodings of the form <start-tag> value <end-tag>, the purpose is to exress ASN.1 data in the XML language. It can be used when moving the data between systems and e.g. databases. XER stands for the XML Encoding Rules.

1.5 A-11 Decode the following: 3A 82 00 10 1A 01 73 1A 02 65 63 1A 81 06 75 72 69 74 79 21. What encoding rule is it?

The 3A in the beginning tells us it is a VisibleString split into three parts and 82 00 10 tells us the that length of the whole string is encoded in the long definite form. We are splitting the string into three parts that are not equal in length. The first part 1A 01 73 tells us that it is a visible string (1A) and the length is 1 (01) and 73 is decoded in ASCII which gives us the letter ”s”. The second part is also a visible string (1A) with the length of 2 (02) and letters ”ec”. The third and last part denotes a visible string (1A) and (81) tells us this part of the string is encoded in the long definite form and the length of the value is 6 (06) which gives us 6 letters. The letters are decoded as ”urity!” and all in all the string is decoded as ”security!”. The encoding rule that was used is known as BER.

1.6 A-16 In CMS, there is a Signed-Data type as well as an Encrypted-Data type. Intriguingly, there is no Signed-Encrypted-Data type, altough it is no doubt useful to encrypt and sign data (at the same time). Why is this?

It might be useful to encrypt and sign the data at the same time since it could save some time but unfortunately there are pitfalls one must avoid. For example if one would first encrypt and then sign, in that case the digital signature does not prove that the sender was aware of the context of the plaintext. In other words, anybody can verify the authenticity and only receiver can decrypt it. Vice versa, i.e. sign and then encrypt only the receiver can decrypt and then verify. Basically, if Signed-Encrypted-Data type existed then security issues would most likely occur. Optimally, one should sign-encrypt-sign and thus if Signed-Encrypted-Data type was available an inexperienced user could use it and supposedly thinking it was secure when it is not. It should be altogether avoided and instead one shuld encrypt and sign the data separately.

1.7 A-17 For signed-data in CMS, several signers can sign the same data. How is this feature achieved?

As mentioned, one useful feature in the signed-data type in CMS is that it allows several signers to sign the same data. In order to achieve this, each of the signer has to prove to the receiver that their certificate is valid. Therefore, information for each signer is provided in the type SignerInfo and is send to the receiver for verification. The contents of SignerInfo includes for example which version of the SignerIdentifier is used and which certificate should be used to verify the signature. Essentially the type consists of information that proves the signer has not any malicious intent against the receiver.

1.8 A-22 In PKCS #12, the PFX type has an optional MacData type used for symmetric data integrity protection. Why is there no optional signature type used for the asymmetric data integrity protection?

There is no optional signature type used for the asymmetric data integrity protection since in public key integrity mode of the asymmetric data integrity protection uses digital signatures for protection. Only in password integrity mode of the symmetric data integrity protection the MAC together with a key derived from a password is used thus it explains why MacData type is exclusively used for symmetric and not asymmetric (also known as public key).