1 ASN.1

You are to define a protocol for communication between an automatic scale and a packing machine.

1. The scale measures the weight in grams as a floating point number and the code number of the merchandise as an integer. Define an ASN.1 data type ScaleReading which the scale can use to report this to the packing machine.
2. Some countries use, as an alternative to the metric system, a measurement system based on inches, feet and yards. Define a data type Measurement which gives one value in this system, and Box which gives the height, length and width of an object in this measurement system. Feet and yards are integers, inches is a decimal value.
3. Change the definition of Measurement in Exercise 2 so that feet can only have the values 0, 1 or 2 (since 3 feet will be a yard), and so that inches is specified as an integer between 0 and 1199 giving the value in hundreds of an inch (since 1200 or 12 inces will be a foot).

(1) ScaleReading ::= SEQUENCE {

    weight REAL,
    barcode INTEGER
}

Box ::= SEQUENCE {
    height MEASUREMENT
    length MEASUREMENT
    width MEASUREMENT
}

(2) + (3)

Measurment ::= SEQUENCE {

    inches INTEGER,
    feet INTEGER (0..2),
    centinches INTEGER (0..1199)

EXAM

2 Certificates

1. What is the major issue solved by Public Key Infrastructures?
2. What is a X.509 certificate and what are the informations contained in it?
3. Why publish electronic certifcates in a repository?
4. Why publish revocation lists in a repository?
5. From X.509 version 2, one can find the following information inside an X.509 certificate:

```
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
}
```

   what is the use of the `issuerUniqueID` and `subjectUniqueID` fields?
6. A certificate repository is accessible via LDAP and can return X509 certificates. Suppose that an entity Alice wishes to control the validity of the signature of a document signed by an entity Bob. Alice knows the identity of Bob and, say, the repository address. List the sequence of about 5 operations that Alice must realize to check the signature of Bob.
7. Discuss about the following scenarios in terms of security:
   (a) Two distinct certificates are signed with the same private key.
   (b) Two distinct certificates contain the same public key.
   (c) Two distinct certificates have the same subject.
   (d) Two distinct certificates have the same serial number.
   (e) Two distinct certificates have the same issuer.
   (f) Two distinct certificates have the same signature.

(1)

The main issue solved by PKI is to support the architecture of certificates that allow to guarantee the link between the identity and the public key.

(2)

The main components of an X.509 certificates are

Identifier, Validity, Public key, Signature by CA

(3)

The need of publishing certificates in a repository is related to the need of efficently sharing them and allowing one user to recover them if it needs.

(4)

So that each user is able to retrive the list of revoked certificates. This also gives the possibility to a user to verify if the certificates that he receved is still valid or not (or if at some point in the past it was valid).

(5)

The issuerUniqueID and subjectUniqueID are old identifiers for certificates that have been kept for compatibility reasons. This is why they are optional fields and they represent the following:

→ issuerUniqueID: unique identifier given to each CA

→ subjectUniqueID: identify the entity whose public key the certificate binds to.

(6)

Alice may do the following:

- request the certificate to the LDAP
- receives the certificate and verifies that the identity-public key matches
- query OSCP server and gets back the result
- check if the signature on the document is correct

(7)

a → No problem, CA syn multiple cert with the same key

b → This means that two certificate have the same key and may have been valid in different moment. Or one of the two is a malicious one.

c → This may be possible, a single org may have multiple certs.

d → It means that two issuers have produced the same serial number 😱

e → No problem, it happens all the time

f → Huge problem → They must not exist together 😱

3 Key recovery

1. In a Public Key Infrastructure, who generates the public and private keys?
2. Some PKI infrastructures propose key recovery mechanisms in case of lost keys. What does that implies on the cryptographic services provided by different types of keys (encryption keys versus signature keys)?
3. Even without key recovery agents, a private key could be compromised (that's what revocation list are for). Without which mechanism non-repudiation would then be completely nullified?

(1)

In the PUBLIC KEY INFRASTRUCTURE the users should be able to generate the keys (offline mode)

(2)

If there is the possibility to recover the key it means that the key is held by someone else (generally) so we may loose non-repudiation.

(3)

The non-repudiation holds true if the key is kept secret. The non-repudiation could be lost in case of key recovery if the key is not encrypted.

The United Nations specialized agency for information and communication technologies, the ITU (International Telecommunication Union), has proposed the following modification of the authentication framework for PKI certificates:

```
ENCRYPTED{ToBeEnciphered} ::= BIT STRING

HASH{ToBeHashed} ::= SEQUENCE {
  algorithmIdentifier  AlgorithmIdentifier{{SupportedAlgorithms}},
  hashValue            BIT STRING
}

ENCRYPTED-HASH{ToBeSigned} ::= BIT STRING

SIGNATURE{ToBeSigned} ::= SEQUENCE {
  algorithmIdentifier  AlgorithmIdentifier{{SupportedAlgorithms}},
  encrypted            ENCRYPTED-HASH{ToBeSigned}
}

SIGNED{ToBeSigned} ::= SEQUENCE {
  toBeSigned  ToBeSigned,
  COMPONENTS OF SIGNATURE{ToBeSigned}
}

Certificate ::= SIGNED{ TBSCertificate }
```

1. What should be the bit string in ENCRYPTED?
2. What should be the bit string in HASH?
3. What should be the bit string in ENCRYPTED-HASH?
4. What can be the meaning of the constructor COMPONENTS OF? Then compare the proposed Certificate construct to the X.509 reference:

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

①

This should be the result of applying an encryption function to the toBeEncrypted bytes

②

This should be the result of applying a crypto hash function to the toBeEncrypted bytes

③

It should be the result of applying ① to the output of ②

④

The sequence is replaced by the list of the element of the SEQUENCE in which it is wrapped.

5 CRL extensions

Q1. What is the main usage of Certificate Revocation Lists and what are the contents of a CRL published by a PKIX-like hierarchical PKI?
Q2. What is the effect of revocation on the lifetime of certificates?
We are now going to study some possible extensions of CRLs.

(Q1)

CRLs are used to revoke certificates that are no longer valid. This is to allow users to check if a certificate is still valid or not. The content may be as follows:

- CRL header information
- Revoked certificate list
- CRL extensions
- Signature

(Q2)

The lifetime of a certificate is mainly determined by the validity of the certificate itself. However, it can be shorter is it appears in a CRL

5.1 CRL Issuing Distribution Point

Excerpt of the [RFC5280]:
[...] *The issuing distribution point is a* critical CRL extension *that identifies the CRL distribution point and scope for a particular CRL. It indicates whether the CRL covers revocation for end entity certificates only, CA certificates only, attribute certificates only, or a limited set of reason codes.* [...]

```
IssuingDistributionPoint ::= SEQUENCE {
     distributionPoint          [0] DistributionPointName OPTIONAL,
     onlyContainsUserCerts      [1] BOOLEAN DEFAULT FALSE,
     onlyContainsCACerts        [2] BOOLEAN DEFAULT FALSE,
     onlySomeReasons            [3] ReasonFlags OPTIONAL,
     indirectCRL                [4] BOOLEAN DEFAULT FALSE,
     onlyContainsAttributeCerts [5] BOOLEAN DEFAULT FALSE }

     -- at most one of onlyContainsUserCerts, onlyContainsCACerts,
     -- and onlyContainsAttributeCerts may be set to TRUE.
```

Q1. Give some examples of reason codes.
Q2. What could be the reason of using this extension for a CRL?
Q3. What can be the risk of using this subsets of CRLs?

(Q1)

An example could be "private_key_compromized" or "identity_missmatch"

(Q2)

This allows to reduce the burden of having hyge CRLs and make the access to CRLs more granular. This means that if someone is interested in specific CRL types he has the possibity to retrieve just that.

(Q3)

We may have a limited visibility and miss some certificate that are not in our orginal query.

5.2 Delta CRL indicator

Excerpt of the [RFC5280]:
[...] *The delta CRL indicator is a* critical CRL extension *that identifies a CRL as being a delta CRL. The delta CRL indicator extension contains the single value of type* `BaseCRLNumber`. *The CRL number identifies the CRL, complete for a given scope, that was used as the starting point in the generation of this delta CRL. A conforming CRL issuer MUST publish the referenced base CRL as a complete CRL. The delta CRL contains all updates to the revocation status for that same scope. The combination of a delta CRL plus the referenced base CRL is equivalent to a complete CRL, for the applicable scope, at the time of publication of the delta CRL.*
When a conforming CRL issuer generates a delta CRL, the delta CRL MUST include a critical delta CRL indicator extension. [...]

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

A complete CRL and a delta CRL MAY be combined if the following four conditions are satisfied:

a. The complete CRL and delta CRL have the same issuer.
b. The complete CRL and delta CRL have the same scope. The two CRLs have the same scope if either of the following conditions are met:
   (i) The issuingDistributionPoint extension is omitted from both the complete CRL and the delta CRL.
   (ii) The issuingDistributionPoint extension is present in both the complete CRL and the delta CRL, and the values for each of the fields in the extensions are the same in both CRLs.
c. The CRL number of the complete CRL is equal to or greater than the `BaseCRLNumber` specified in the delta CRL. That is, the complete CRL contains (at a minimum) all the revocation information held by the referenced base CRL.
d. The CRL number of the complete CRL is less than the CRL number of the delta CRL. That is, the delta CRL follows the complete CRL in the numbering sequence.

[...]
Q1. What is a delta-CRL?
Q2. Explain why when a conforming CRL issuer generates a delta CRL, the delta CRL MUST include a critical delta CRL indicator extension.
Q3. Explain the case a.

Q4. Explain the case bi.
Q5. Explain the case bii.
Q6. Explain the case c.
Q7. Explain the case d.

Q1

Delta CRLs are smaller CRLs compared to the original revocation list. They contain all the certificate from the previous Delta CRL up to now.

Q2

If no idication is given there is no way to know which is the starting point of the delta CRL

Q3

This is because only the issuer of the main CRL can create a complement delta CRL with all relevant informations

Q4

This is because if it is omitted, it means that it has to be applyed to all the scopes. Hence, the two CRLs are compatible

Q5

This is because if the scopes are specified they have to match in order to produce a consistent and coherent result

Q6

We have to assure that no information is lost when combining the two so we have to check that the delta CRL is subsequent to the CRL that it complets.

Q7

The delta CRL contains information to the subsequent revoked certs.

5.3 Indirect CRL

Indirect CRL: in principle, CRL's are emitted by a *CRL issuer*. Generally, the CA itself plays this role as it has to publish CRL's to provide up to date information on its certificates. A CA can nonetheless delegate this task to another trusted party. Then when this CRL issuer is distinct from the CA, the CRL is said to be *indirect* and `indirectCRL` field in the CRL extension, must be set to `true`. Who has to sign an indirect CRL and how to check its validity?

The indirect CRL is signed by the trusted issuer and so to check its validity we have to verify the issuer certificate. This means that we may also verify the link between the original CA and the new issuer.

6 Key Identifier

The Authority Key Identifier (AKID) and Subject Key Identifier (SKID) are certificate extensions that can be used to help facilitate the certification path construction process. As discussed in X.509 and the Internet Certificate and CRL Profile [RFC3280], AKIDs are used to distinguish one public key from another when a given Certification Authority (CA) has multiple signing keys, and SKIDs provide a means to identify certificates that contain a specific public key.

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

There are multiple methods available for calculating the keyIdentifier. The following methods are specifically mentioned in the Internet Certificate and CRL Profile:

- the 20 byte SHA-1 value of the subject public key (not including tag, length, and unused bits)
- a four bit value 0100 followed by the least significant 60bits of the SHA-1 value of the subject public key (not including tag, length, and unused bits)
- a monotonically increasing sequence of integers

1. For these three functions, what can you say about the unicity of the key identifiers in a multiple PKI domains setting? Quantify your answer.
2. Propose a certificate renewal policy and briefly justify it.
3. What about SKID unicity when there are certificate renewals?
4. How is it possible to guaranty the selection of one certificate over another?

(1)

For what concerns the first method:

→ We generate a collision if we have two equal subject in two CA domains

For what concerns the second method:

→ Here collisions depend on the collision of SHA-1 (which is not the best). Since all Kpub would be different the only source of problem may be the hashing algorithm

For what concerns the third method:

→ Here we would need coordination among CA, maybe slicing the intervals for each one of them

(2)

Among the options we have: - not allowing any renewal
- allowing renewal over a small period of time

(3) The same SDKI is kept after renewal since the two certificates have the same public key