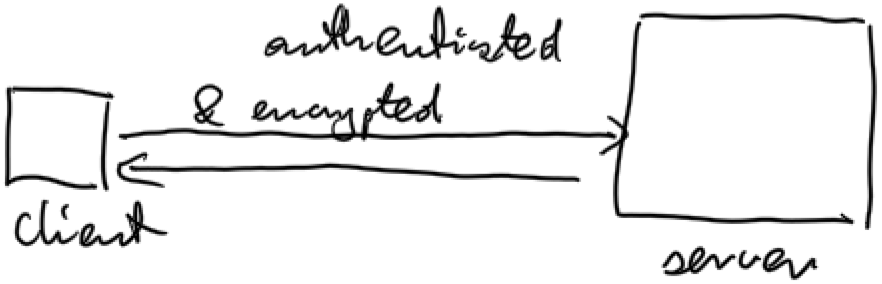
Server authentication:



The server should:

- authenticate:
- ~~make the~~ establish the master session key for encryption and message auth.

The server uses asymmetric crypto:

- DH for key establishment
- signatures: server signs the half of DH.

The (verification) public key must be binded to server's identity

- certificate to used for that purpose

The cert. contains at least:

tbsCertificate

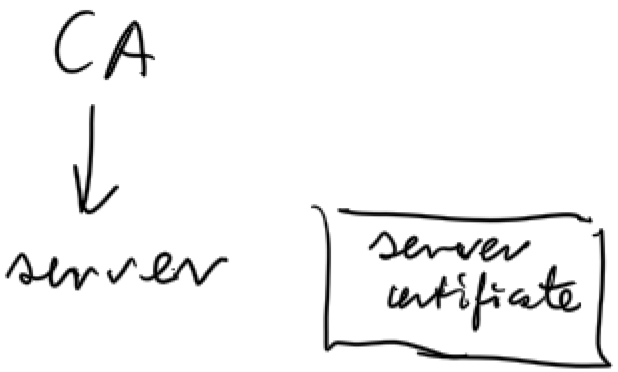
- the name of the issuer
- the name of the subject
- validity period,
- public key description
- information about the purpose of the public key:
  - for signature?
  - for encryption?
  - for certificate issuance?

- description of the signature algorithm used to sign the certificate.
- the signature under the certificate

Why the client trusts in the certificate?

- self signed certs do not scale well.

So we distinguish separate entities:
Certification Authorities

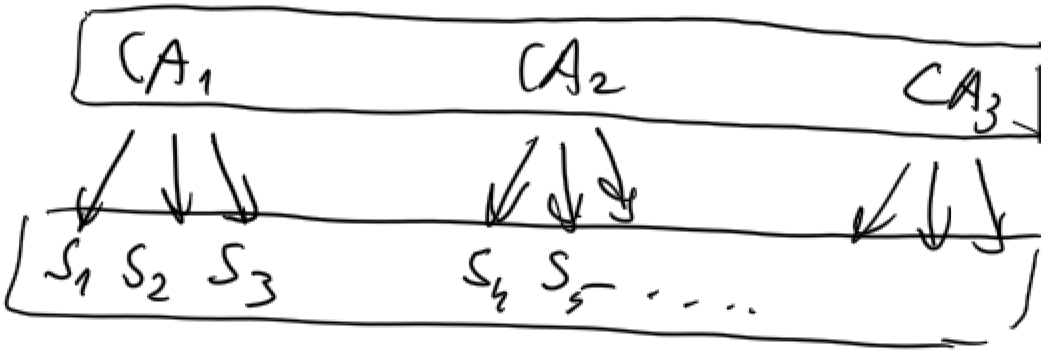


If the client trusts CA then
it trusts in the cert. issued by CA.

- It trusts: - distinguished name of the CA

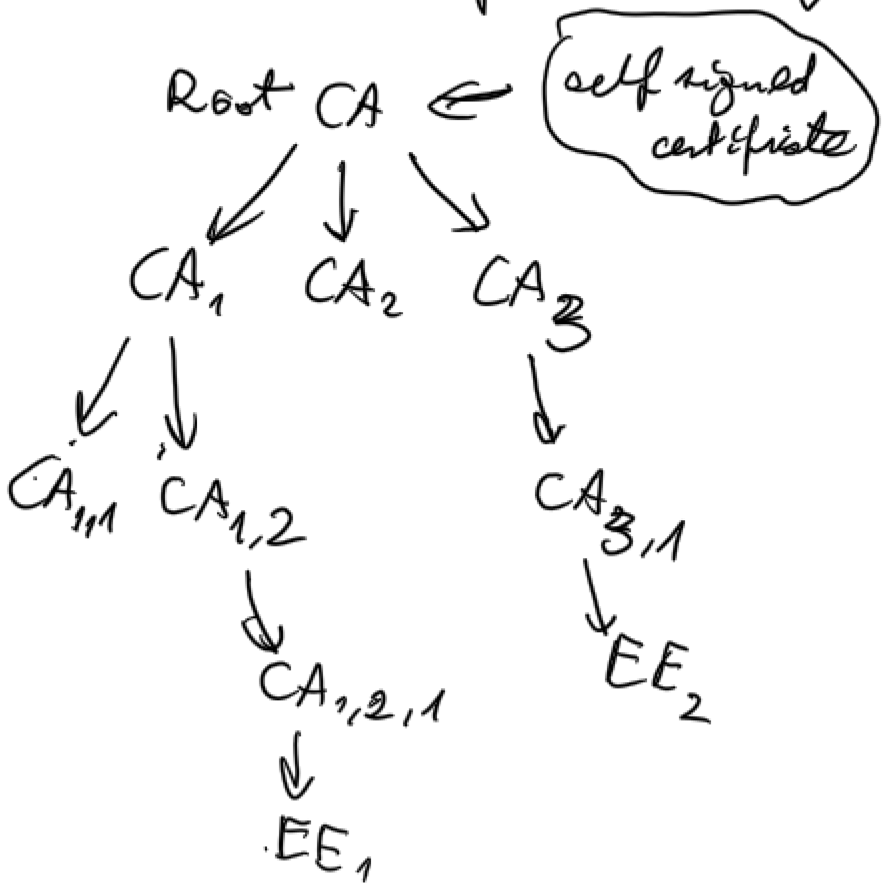
tbsCert.

- knows the verification key to verify cert. issued by CA,
- knows the signature algorithm.



this is not enough for real application.

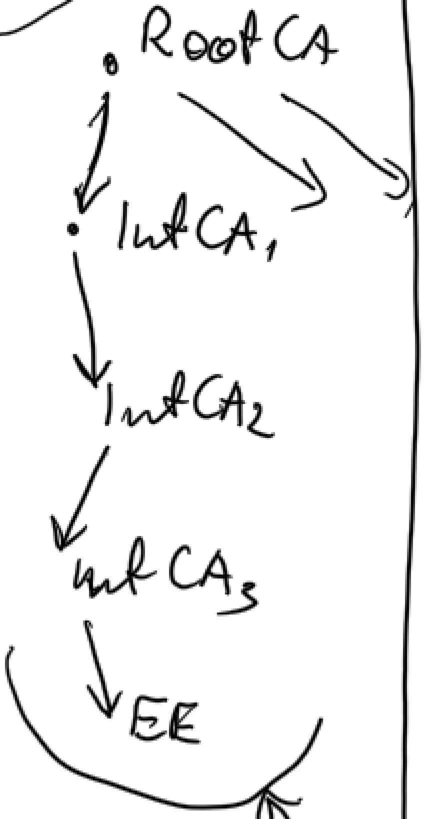
We need a deeper hierarchy:



So the server provides certificate chain:

RootCA, CerIntCA$_1$, CertIntCA$_2$, CertIntCA$_3$, CertEE

the whole chain is sent by the EE (server) to the client.



On one side we have the infrastructure of CAs on the other the certificate chain that reflects the hierarchy.

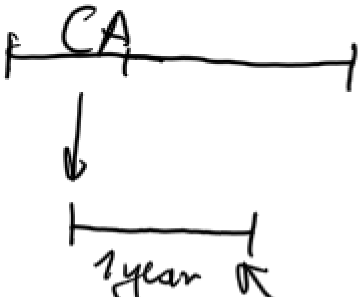
we transfer trust from the top to the bottom of the hierarchy.

Things are not so simple :)

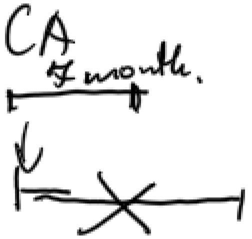
In the certificate we have

Validity field:

NotBefor
NotAfter

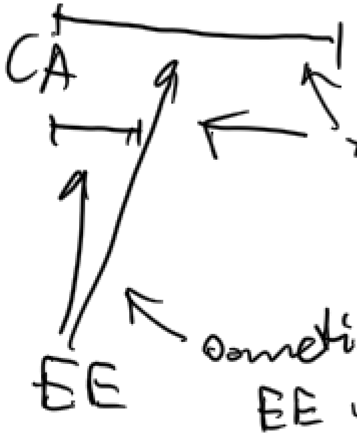


validity period of the cert. used to verify that certificate must include validity period of that certificate



when the validity period of the "superior" key is shorter than for issued certs.

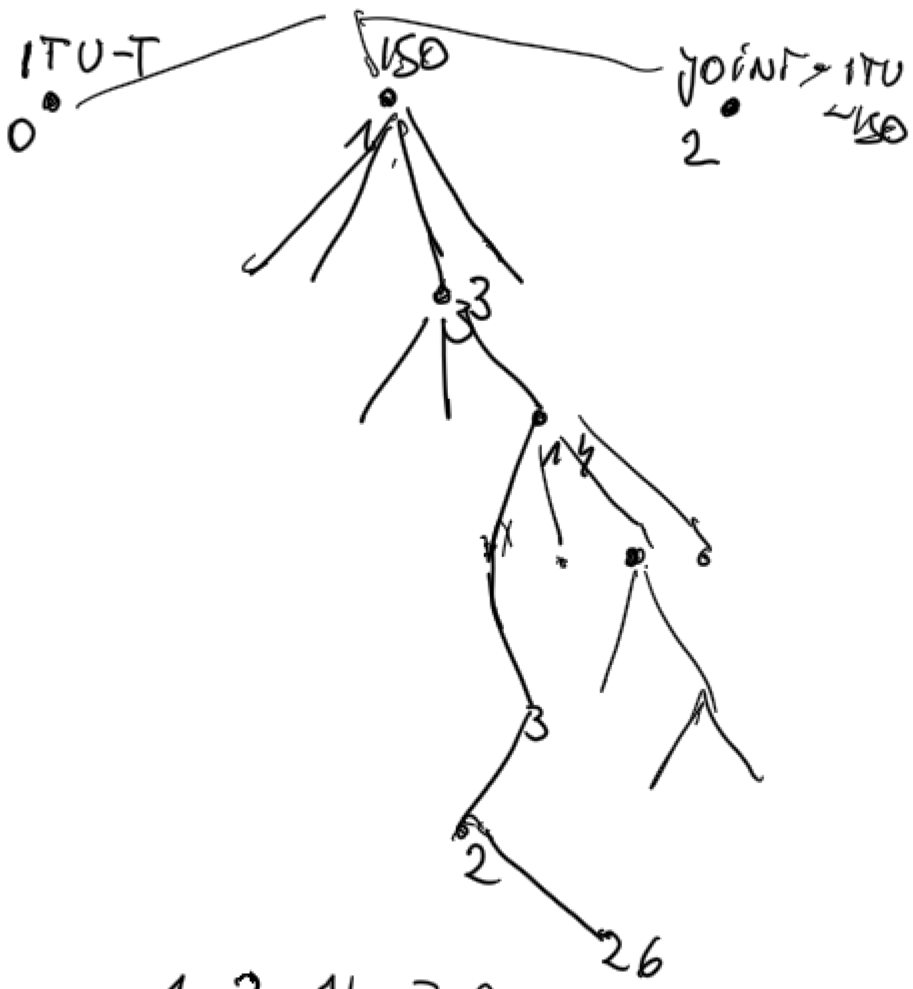
the CA must request a new certificate in the supervising CA, to get a new certificate with long validity period:



two certificates of CA still valid

sometimes EE must indicate which of the public keys (certs) of CA to use to build cert. chain.

So we may follow key identifiers (together with DN of the subjects and Issuers to build the correct chain of certificates).



1.3.14.3.2.26