**Course**: 67515
**My Name**: Yiftach Sabag

# Exercise 2. PKI System

## 1 Implementation Details

### 1.1 Certificate

**Certificate Instance.** The Certificate class represents an issued certificate instance. A certificate has the following fields:

1. Issuer: The entity who issued the certificate.
2. Subject: The the subject who the certificated was issued for.
3. Public key: Public key of the subject.
4. Not before: The time and date which before, the certificate is not valid.
5. Not after: The time and date which after, the certificate is no longer valid.
6. Alternate subject name: domain names which this certificate holds for.
7. Signature: signature signed by the issuer. The signature is signed over the hash content of the certificate
8. CA permission: used to indicate if the subject has CA permissions

**Validity.** A certificate is valid only if the signature can be satisfied with the issuer's public key, and the date is between the range (not before, not after). If one of the certificate's fields is changed, then the issuer's public key will not satisfy the signature. This, way, no one unauthorized can manipulate the certificate.

### 1.2 Entity

**Entity class.** This class represents an abstract entity in the network. this is a super class of the network servers entities. this could be a root CA, CA, Subject.

**Issuing Certificate.** An entity with CA permission can issue a certificate for a Subject instance.

**Revoking Certificate.** An entity with CA permission, can revoke certificate it issued, even if the validity range is not over. A certificate can only be revoked by the issuer.

**Revocation List.** A client can receive from an entity the revoked certificates it issued for different Subject instances.

### 1.3 Certificate Authority

**CA instance.** This entity represents a root CA in the network. When a CA instance is initiated, a self-signed certificate is created for the CA. Thus, is considered a root CA.

### 1.4 Subject

**Subject instance.** The Subject entity represents a server a Client instance can connect to and receive data from.

**Receiving Certificate.** A subject can receive certificates from root CA's or subjects with CA permission.

### 1.5 Client

**Client Instance.** Represents a Client instance in the network, that can store locally certificates of root CA's it knows, and certificates of subjects it receives for different servers.

**Connecting to a server.** A Client can connect to different servers. The connection process follows the simplified-version of TLS protocol shown in class. When receiving a certificate for a server, the client validates the certificate up the Trust-chain, until reaching a root CA, or a subject with CA permissions the client already knows. After the "TLS-handshake", the client can receive data from the server, using symmetric encryption.

**Update.** The client can be updated from an entity regarding the revoked certificates it issued.

### 1.6 Limitations

1. A Client is not automatically updated regarding the manual revocation of certificates.
2. A Subject can have multiple certificates, but it returns the last certificate it received.

## 2 Usage

To use a simple example of the program, via terminal, run the command: `python3 main.py` from root folder directory of the project (where `main.py` is located).