Mosquitto

controllo degli accessi

Configurazione di mosquitto

mosquitto ha molte possibilità di configurazione: per modificare i valori di default è possibile agire sul file

mosquitto.conf

mosquitto cerca questo file in una cartella di default, ma è possibile in fase di avvio indicare quale usare

mosquitto –c mosquitto.conf_mia

Il file di configurazione può contenere degli #include <dir> , in questo caso mosquitto procede ad esaminare i files inclusi in tali directory come se fosse un unico file che li concatena tutti.

Introduzione di autenticazione utenti

Il file mosquitto.conf può essere configurato per limitare gli accessi al broker:

- Permettere la connessione solo ad utenti accreditati
- Limitare i topic accessibili a determinati utenti

Limitazione connessione ad utenti autenticati

mosquitto.conf (sezione Security)

allow_anonymous false

password_file C:\mosquitto\provapass.txt

Il file delle password deve essere preparato: scrivere in un file di testo

utente1:passwU1

utente2:passwU2

Cifrare le password con il comando mosquitto_passwd –U filepass.txt

NOTA: dettagli su https://mosquitto.org/man/mosquitto-conf-5.html

Attivare mosquitto con autenticazione utenti

mosquitto -c mosquitto.conf –v

Se proviamo a connettere il client (publisher o subscriber che sia) senza specificare un utente valido viene rifiutata la connessione

mosquitto_sub -h localhost -p 1883 -t temperatura

ERRORE: RIFIUTATA LA CONNESSIONE

mosquitto_sub -h localhost -u user1 -P passwU1 -p 1883 -t temperatura

mosquitto_pub -h localhost -u user2 -P passwU2 -p 1883 -t temperatura –m 23

Estendere il programma java visto a lezione

```
String password = "passwU1";
char pwd[] = password.toCharArray();
    MqttConnectOptions options = new MqttConnectOptions();
    options.setUserName("user1");
    options.setPassword(pwd);
```

Controllo sui topic

E’ anche possibile limitare l’accesso ai singoli topic definendo una access control list: in mosquitto.conf inserire il nome del file in corrispondenza del parametro acl_file

acl_file C:\mosquitto\acl.txt

user1 può leggere o scrivere qualsiasi topic

user user1

topic readwrite #

user2 può leggere/scrivere solo topic home/temperatura

user user2 (readwrite si può omettere)

topic home/temperature

#user3 può solo leggere la temperatura

user user3

topic read home/temperature

Utilizzare lo username per indicare il topic

È possibile usare lo username all’interno dei topic permessi

home/%u/att

Se la definizione gerarchica dei topic è fatta in modo da distinguere le competenze specifiche di un certo user includendo il suo nome nella composizione del topic stesso

Naturalmente possiamo usare anche + e # secondo la stessa logica dell’uso di queste wildcard in fase di sottoscrizione

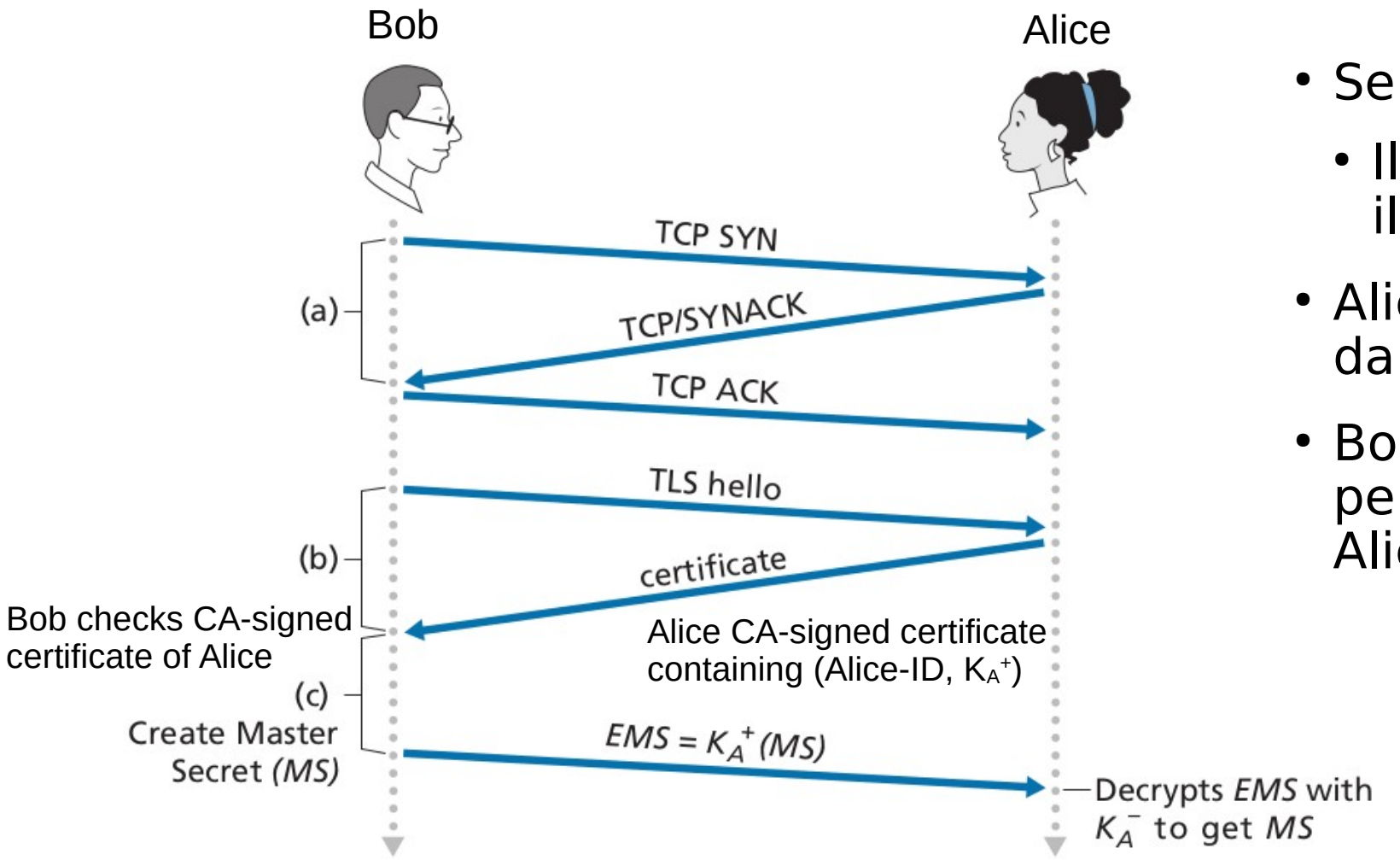
Mosquitto su TLS

Per usare i canali cifrati occorre disporre dei necessari certificati e chiavi ed inoltre bisogna configurare opportunamente mosquitto (file `mosquitto.conf`)

Usiamo openssl per generare certificati e chiavi

Si veda `man mosquitto-tls` per istruzioni dettagliate o `https://mosquitto.org/man/mosquitto-tls-7.html`

Handshake TLS



- Server-side TSL authentication
  - Il più usato, si autentica solo il server
- Alice manda certificato firmato da CA
- Bob usa il certificato della CA per verificare il certificato di Alice

Generare il certificato e le chiavi

- Creare la coppia di chiavi per la nostra CA
- Creare il certificato della CA firmato con la chiave privata della CA
- Creare una coppia di chiavi per il broker
- Creare il certificato del broker e firmarlo con la chiave della CA

Nel file `mosquitto.conf` occorre specificare i pathname dove si trovano i certificati:

```
cafile

certfile

keyfile
```

La nostra Certification Authority

È necessario disporre del certificato della CA (la nostra) che ha firmato il certificato del broker.

Inoltre il broker deve avere un suo certificato, garantito dalla CA .

Il broker deve anche avere una chiave privata per cifrare i propri messaggi

Creazione del certificato e della chiave della CA

```
openssl req -new -x509 -days 600 -extensions v3_req  -keyout
ca.key -out ca.crt -config reqCA.conf  --verbose
```

Si devono inserire diversi dati contenuti nel file `reqCA.conf` e una password (inseriamo pissir),

Ottengo `ca.crt` e `ca.key`

Contenuto ReqCA.conf

```
[req]
distinguished_name = req_distinguished_name
req_extensions = v3_req
prompt = no

[req_distinguished_name]
C = IT
ST = Vercelli
L = VC
O = UPO
OU = Rettorato # distinguished name of server must not match the distinguished name of CA
CN = localhost
[v3_req]
keyUsage = keyEncipherment, dataEncipherment, keyCertSign # keyCertSign to sign server cert
basicConstraints=CA:TRUE # specify that it is a CA
```

Preparare il certificato per il server

Nel nostro caso la CA è «in casa», comunque il comando è:

```
openssl genrsa -out server.key 2048
```

Ora abbiamo la server.key e dobbiamo farla firmare dalla CA.

```
openssl req -out server.csr -key server.key  --new -config
reqServ.conf --verbose
```

Nuovamente I dati del server sono presenti nel file `reqServ.conf`.

Attenzione a dare come common name il domain name del server (uso `localhost`)

A questo punto abbiamo `server.csr` e `server.key` e dobbiamo usare la chiave della CA per “firmare” il certificato del server.

Contenuto reqServ.conf

```
[req]
distinguished_name = req_distinguished_name
req_extensions = v3_req
prompt = no

[req_distinguished_name]
C = IT
ST = Vercelli
L = VC
O = UPO
OU = DISIT # distinguished name of server must not match the distinguished name of CA
CN = localhost
[v3_req]
keyUsage = keyEncipherment, dataEncipherment
```

Firmare il certificato del server con la chiave della CA

```
openssl x509 -req -in server.csr -CA ca.crt -CAkey ca.key -
CAcreateserial -out server.crt -days 500
```

(l'ultimo parametro dice per quanti giorni varrà il certificato).

Output:

```
Signature ok
subject=C = IT, ST = Vercelli, L = VC, O = UPO, OU = DISIT,
CN = localhost
Getting CA Private Key
Enter pass phrase for ca.key:
```

<<< inserire la passphrase: pissir

Ora abbiamo anche server.crt

Configurare il broker

In `mosquitto.conf` dobbiamo aggiungere

1) Il listener sulla porta 8883
2) Dove trovare i file con i certificati impostando i parametri:

`cafile` – certificato della certification authority

`certfile` – certificato del broker

`keyfile` – chiave del broker

Connessione sulla porta 8883

```
./mosquitto_sub -h localhost -p 8883 -t
pissir/prova --cafile pathname_file_ca.crt
```

```
./mosquitto_pub  -h localhost -p 8883 -t
pissir/prova -m "ciao Ciao“ --cafile
pathname_file_ca.crt
```

L’hostname deve essere identico al Common Name (CN) del certificato, altrimenti la connessione viene rifiutata!