Application layer: SSH

* Applicatielaag maar bevat ook deel bovenop de transportlaag zodat andere applicatielaag protocols veilig (encrypted) kunnen praten met de transportlaag
* Beveiliging voor connectie andere host over het netwerk
* SSH1 voor remote login in, SSH2 ook meer shit
* Ook tunneling, file transfer (scp), mogelijk
* Zeer hoge encryptie bits mogelijk maar groot netwerkbelasting hierdoor

# Architectuur

## SSH transport layer protocol

* server authenticeit, integriteit, confidentialiteit + compression (optioneel) + boven betrouwbaar TCP
* Server authenticatie gebeurt dus op transport laag gebaseerd op public-private sleutelpaar

Twee mogelijke trust models

1. Client heeft lokale db die hosts aan een publieke host sleutel koppelt
2. Host name to key associatie gebeurd door een CA

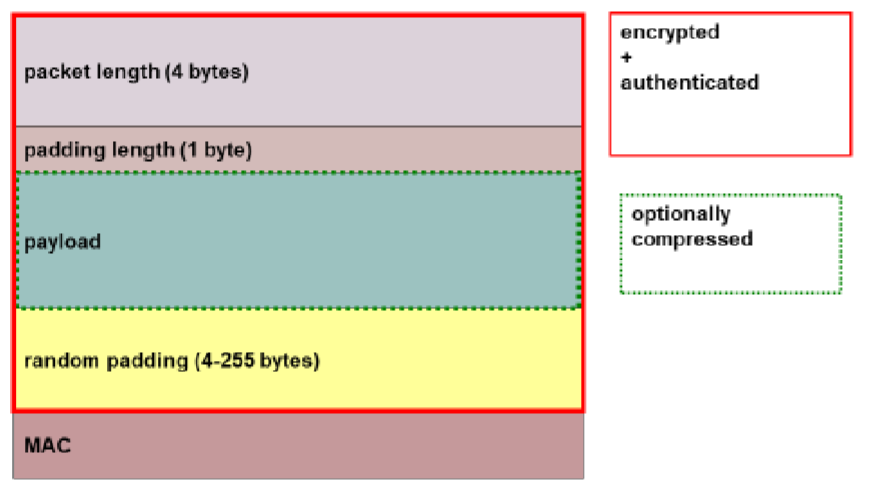
#### Security algorithm negotiation

4 stappen

1. Identificatie string uitwisseling: ”SSH-protoversion-softwareversion SP comments CR LF”
2. Algorithm negotiation
   * elk stuurt SSH\_MSG\_KEXINIT (lijst supported algoritmes)
   * voor key exchange, mac adres, encryption en compression
   * wordt telkens het eerste algoritme van de client gekozen dat door de server ondersteunt wordt
3. 2 versies van diffie-hellman (asym.) ondersteunt
4. Service request (SSH\_MSG\_SERVICE\_REQUEST) wordt door client gestuurd naar authenticatie- of connectie-protocol.

Na deze stappen wordt alles verstuurd als SSH transport layer packet.

#### Binary packet protocol



* Pakket lengte: van alles behalve pakketlengteveld en MACveld

### Key exchange

* Per sessie

Server stuurt: shared session key (diffie), host public key, signature

Server authentication: op basis server host key  
key identificiation via: local db’s, CA’s, offline channels, best effort approach (eerste keer daje connect het in lokale db steken)

Key re-exchange: kan elk moment door elke partij, sessie ID blijft zelfde maar ander algoritme eventueel, session key anders. Doeje best na X aantal tijd/data.

## SSH user authentication protocol

* client authenticatie (met server), bovenop SSH transport layer

-Public key: client stuurt zijn public key signed met zijn private key.   
-Password  
-Host based: authenticatie op basis van de host van de client

## SSH connection protocol

* secure tunnel + user authentication = logical channels

channels zijn: sessions, x11 connecties, local/remote port forwarding

* Multiplexed in enkele encrypted tunnel
* Channel nummer op elk eind connectie

#### Local port forwarding

#### Remote port forwarding