UiO : Institutt for informatikk

# Gruppetime IN1020 – Uke 10

## Datanettverk

*Erling Holte*
*erlinhol@uio.no*

# Plan for timen

- Litt repetisjon av TCP/IP-modellen
- Regneoppgaver
- Praktiske oppgaver på linux-maskinene
- Jobbe med oblig/ukesoppgaver

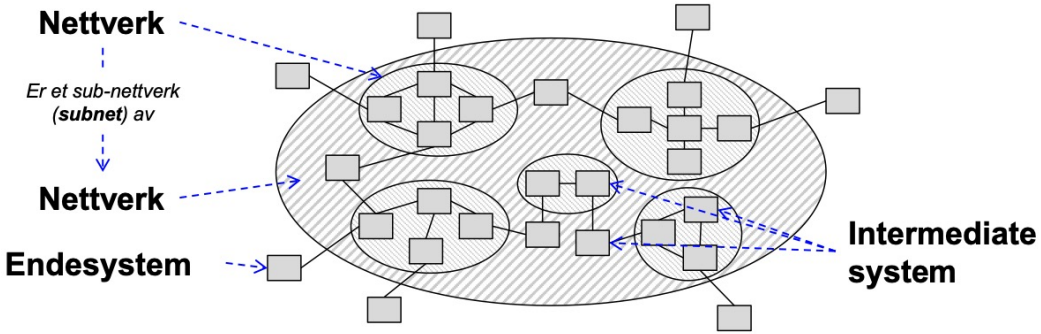
# TCP/IP-modellen enkelt forklart ☺

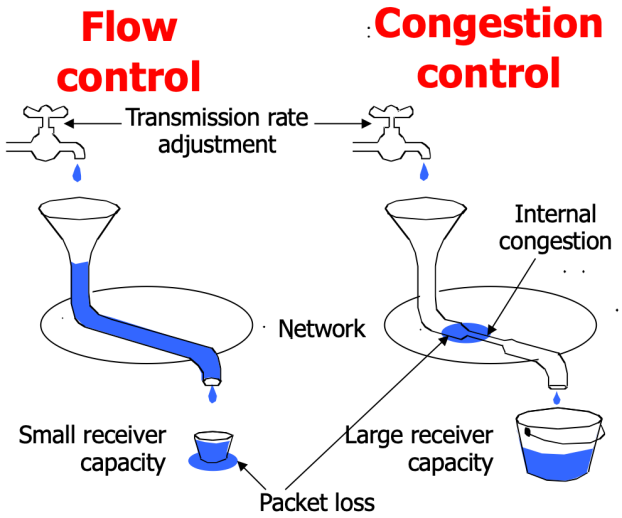
https://www.youtube.com/watch?v=PpsEaqJV_A0&t=1s

# TCP/IP-modellen

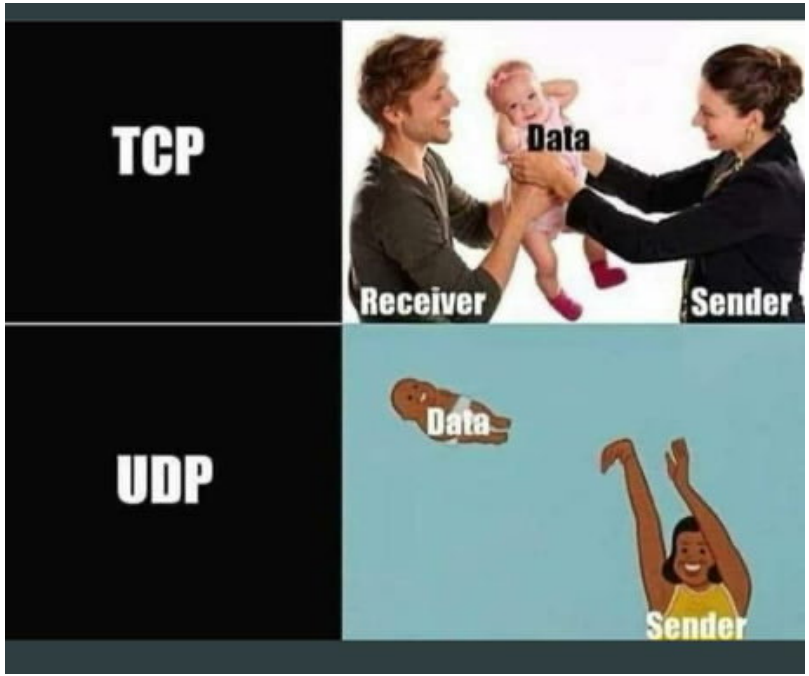
| Lag | | Funksjon |
|---|---|---|
| 5 | **Applikasjon** | Applikasjonsrelaterte tjenester |
| 4 | **Transport** | Kobler sammen systemene ende-til-ende (TCP/UDP) |
| 3 | **Nettverk** | Rute data fra ende-til-ende systemer (IP) |
| 2 | **Link** | Pålitelig overføring mellom to noder |
| 1 | **Fysisk** | Sender bit ut på mediet (kablet eller trådløst) |

# Metningskontroll og flytkontroll



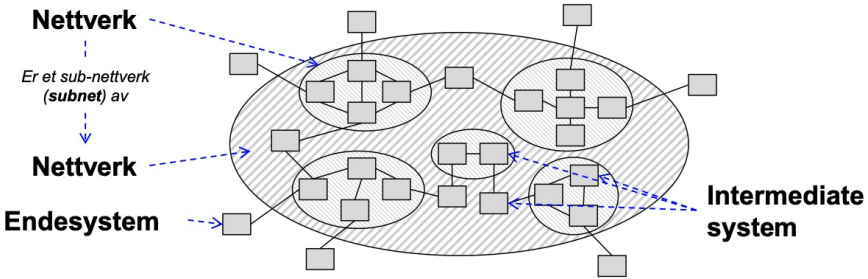



# TCP og UDP



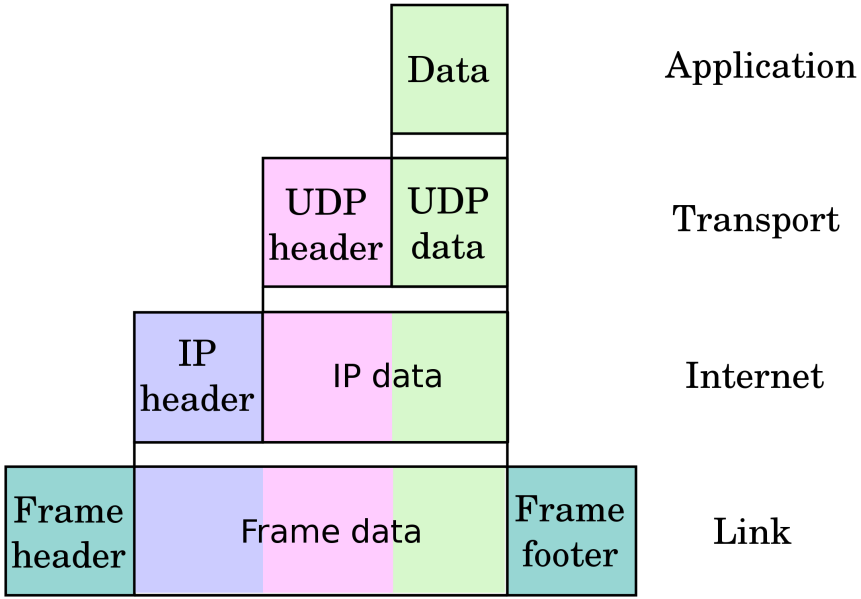

TCP er forbindelsesorientert
Tilbyr metningskontroll og flytkontroll

UDP er tilkoblingsløs
«Best effort» - pakker kan gå tapt

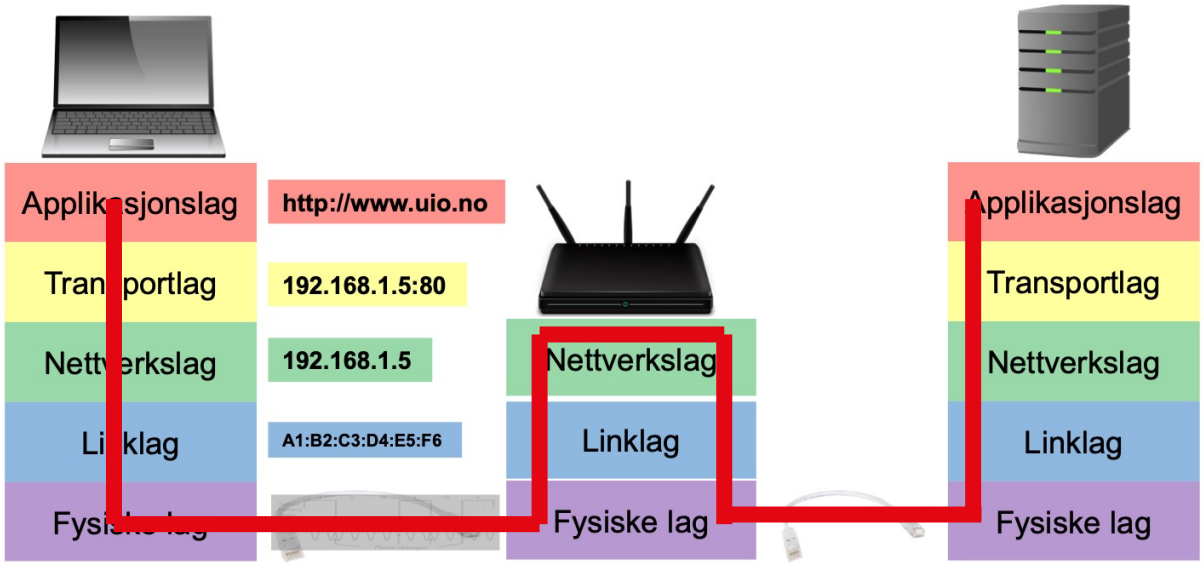



*Hva er fordeler og ulemper med protokollene?*

# Lagene legger på headere



# Hvert lag pakker dataene inn i pakker med nye pakkehoder



# Nettverksmaske

- 0 = vertsdel
- 1 = nettverksdel

255.255.0.0

11111111.11111111.00000000.00000000

- Punktnotasjon må ha med hele denne nettverksmasken
- CIDR-notasjon angir bare hvor mange hvor mange bits som er nettverksdelen

- Nettverksmaske brukes til å dele en IP-adresse i subnettverk. Spesifiserer hva som er tilgjengelig for verter

# Fra 10-tallsystemet til binært

- En IP-adresse består av oktetter
- Vi har gitt IP-adressen ved CIDR-notasjon: 192.168.0.30/24

| 10-tallsystemet | 192 | 168 | 0 | 30 |
|---|---|---|---|---|
| Binært | 11000000 | 10101000 | 00000000 | 00011110 |

- Gjøres om til binært ved å gjøre om hver oktett (altså de som har punktum mellom seg)

# «Regne ut» nettverksmasken fra IP med CIDR notasjon

- Vi har gitt IP-adressen ved CIDR-notasjon: 192.168.0.30/24
- Fordi det står /24 blir det 24 enere i starten: **11111111.11111111.11111111.00000000**
- Eller skrevet i 10-tallsystemet: 255.255.255.000

## Nettverksmaske

- 0 = vertsdel
- 1 = nettverksdel

255.255.0.0

11111111.11111111.00000000.00000000

- Punktnotasjon må ha med hele denne nettverksmasken
- CIDR-notasjon angir bare hvor mange hvor mange bits som er nettverksdelen

- Nettverksmaske brukes til å dele en IP-adresse i subnettverk

# Regne ut subnettet fra en IP og nettverksmaske

- *En bitvis AND operasjon mellom IP-adressen og nettverksmasken*
- Nettverksmaske: 11111111.11111111.11111111.00000000

| IP | 11000000 | 10101000 | 00000000 | 00011110 |
|---|---|---|---|---|
| Nettverksmaske | 11111111 | 11111111 | 11111111 | 00000000 |

**AND**

- Resultat: **11000000.10101000.00000000.00000000**
- Subnettadresse i punknotasjon (binært): 11000000.10101000.00000000.00000000
- Subnettadresse i punknotasjon (10-tallsystemet): 192.168.0.0
- Subnettadresse i CIDR-notasjon: 192.168.0.0/24

# Hvor mange IP-adresser er det i vertsdelen av dette subnettet (fra forrige oppgave)?

- Som vi så tidligere er det 8 bit satt av til **vertsdelen**. Altså skulle en tro at det blir $2^8$=256 adresser. Imidlertid er det **alltid:**
- En adresse til routeren
- En adresse til broadcast
- 256-2 = 254
- Det er 254 adresser i vertsdelen av subnettet

# Regne ut kringkastingsadressen til et subnett

- *En bitvis OR operasjon mellom maskinens **IP-adresse** og nettverksmasken **invers***

| IP | 11000000 | 10101000 | 00000000 | 00011110 |
|---|---|---|---|---|
| Nettverksmaske invers | 00000000 | 00000000 | 00000000 | 11111111 |

- Resultat: **11000000.10101000.00000000.11111111**
- Kringkastingsadresse til subnettet i punktnotasjon (binært): 11000000.10101000.00000000.11111111
- Kringkastingsadresse til subnettet i punktnotasjon (10-tallsystemet): 192.168.0.256
- Kringkastingsadresse til subnettet i CIDR-notasjon: 192.168.0.256/24

# Overføringshastighet (eksamen 2019 – prøveeksamen)

- Du ønsker å laste ned en fil på 200 megabyte, og den maksimale nedlastingshastigheten på din Internettforbindelse er 20 megabit per sekund. Hva er den teoretisk korteste overføringstiden?

- 20 sekunder
- 80 sekunder
- 10 sekunder
- 5 sekunder
- 100 sekunder

$$v = \frac{s}{t} \longrightarrow t = \frac{s}{v}$$

$s = størrelse\ på\ filen$

$v = nedlastningshastighet$

$$t = \frac{200MB}{20Mbit/s} \longrightarrow t = \frac{200MB * 8bit/B}{20Mbit/s} \longrightarrow \underline{t = 80\ s}$$

# DHCP

Dynamic Host Configuration Protocol

- Som oppgave å tildele IP-adresser til nye enheter
- Automatisk utdeling av IP-adresser
- Består av:
  - DHCP discover – ser etter en DHCP-tjener (broadcaster beskjeden)
  - DHCP offer – tilbyr en adresse
  - DHCP request – ber om å få bruke den gitte adressen
  - DHCP ack – gir IP-adressen enheten
- Som regel ruteren som står for utdeling av IP-adresser

# ARP

Address Resolution Protocol

- Kobler internett og linklaget sammen
- Avsenderen må vite hvilken IP-adresse pakken skal sendes til
- En tabell, ARP cache, holder oversikt over korrelasjon mellom IP-adresser og MAC-adresser
- På mange måter likt hvordan DHCP fungerer, men på linklaget

# Porter

| Port | Tjeneste |
|---|---|
| 0 | Reservert |
| 1 | tcpmux |
| … | |
| 22 | SSH |
| … | |
| 80 | HTTP |
| … | |
| 1024-49151 | Brukerporter |
| 49152-65535 | Dynamisk / privat |

En IP-adresse kan ha ulike porter

http://www.something.com:22/MyService

http://www.something.com:80/MyService

# NAT

Network Address Translation

- IPv4 har for få adresser
- NAT er løsningen for hvordan man kan bruke adresser flere ganger
- NAT oversetter fra lokale til offentlige IP-adresser
- Bruker ledige porter til å lage en ny IP-adresse
- Ruteren har en NAT-tabell, som er en oversikt over mappingen

| Kilde IP | Mottaker | Oversatt adresse |
|---|---|---|
| 192.168.1.100 | 212.14.32.173:80 | 46.67.132.46:41254 |
| 192.168.1.200 | 212.14.32.173:80 | 46.67.132.46:41300 |

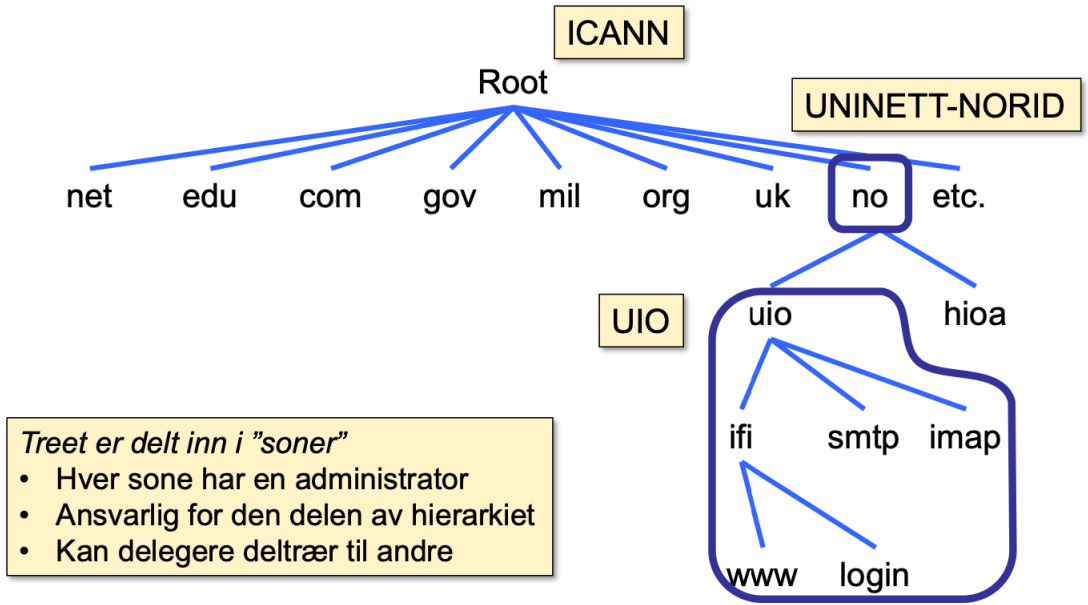
# Private IP-adresser

| RFC1918 name | IP address range | number of addresses | largest CIDR block (subnet mask) | host id size | mask bits |
|---|---|---|---|---|---|
| 24-bit block | 10.0.0.0 – 10.255.255.255 | 16,777,216 | 10.0.0.0/8 (255.0.0.0) | 24 bits | 8 bits |
| 20-bit block | 172.16.0.0 – 172.31.255.255 | 1,048,576 | 172.16.0.0/12 (255.240.0.0) | 20 bits | 12 bits |
| 16-bit block | 192.168.0.0 – 192.168.255.255 | 65,536 | 192.168.0.0/16 (255.255.0.0) | 16 bits | 16 bits |

Disse IP-adressene skal ikke være direkte koblet mot internett

# DNS

Domain Name System




*Treet er delt inn i "soner"*

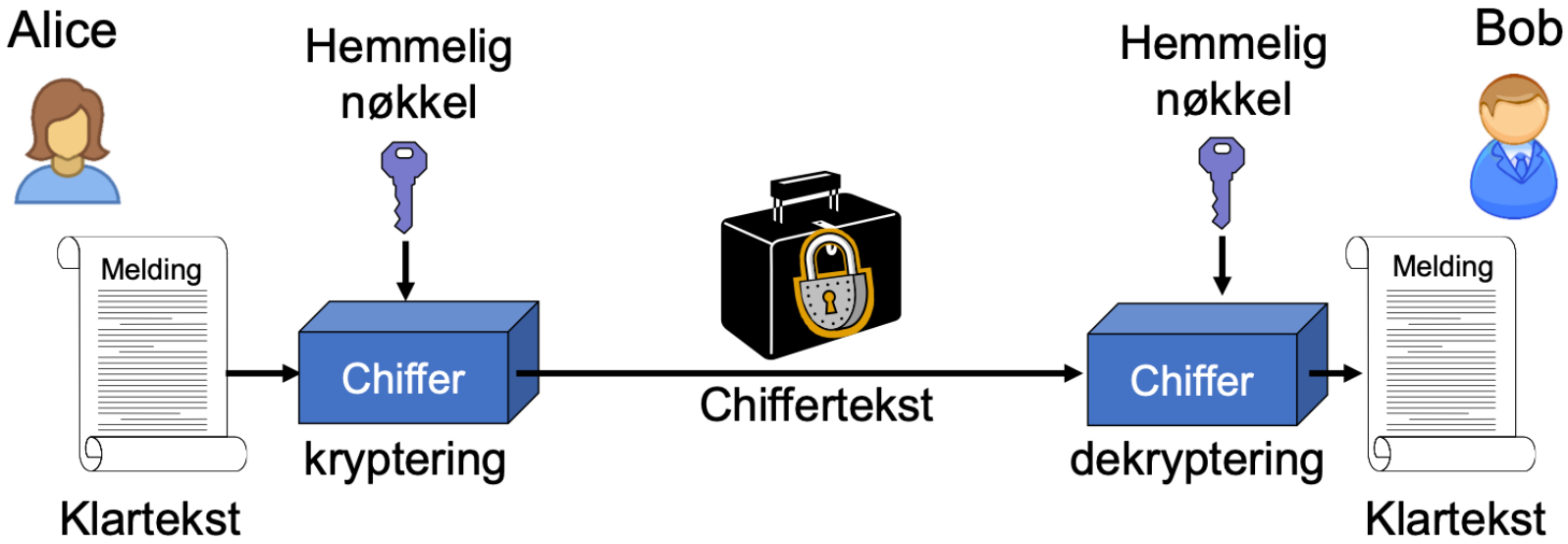
- Hver sone har en administrator
- Ansvarlig for den delen av hierarkiet
- Kan delegere deltrær til andre

# Kryptering

Kryptografi = vitenskapen om hemmelig skrift

- Krypteringsalgoritmer
  - Symmetrisk
  - Asymmetrisk
- Hash-funksjoner
- Kryptering er ikke noe nytt – Cæsar-chifferet er et av de første formene for kryptering
- 2. verdenskrig: Enigma
- Sikkerhetsmål:
  - Konfidensialitet
  - Integritet
  - Autentisitet
  - Uavviselighet

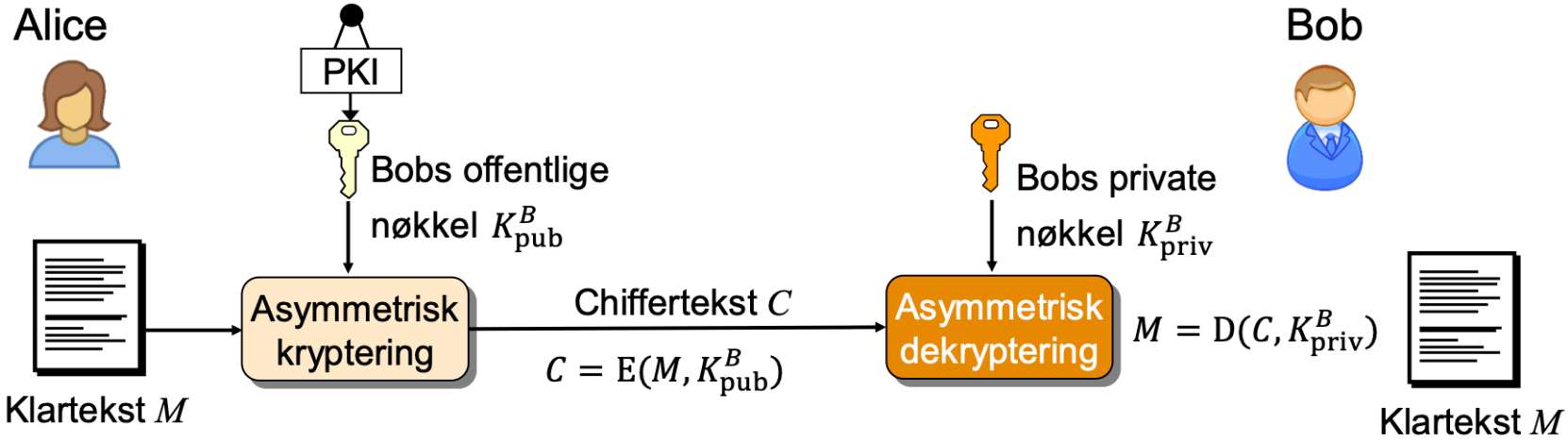
# Symmetrisk kryptering




Gir autentisitet, men ikke uavviselighet

*Illustrasjon hentet fra IN2120 Informasjonssikkerhet*

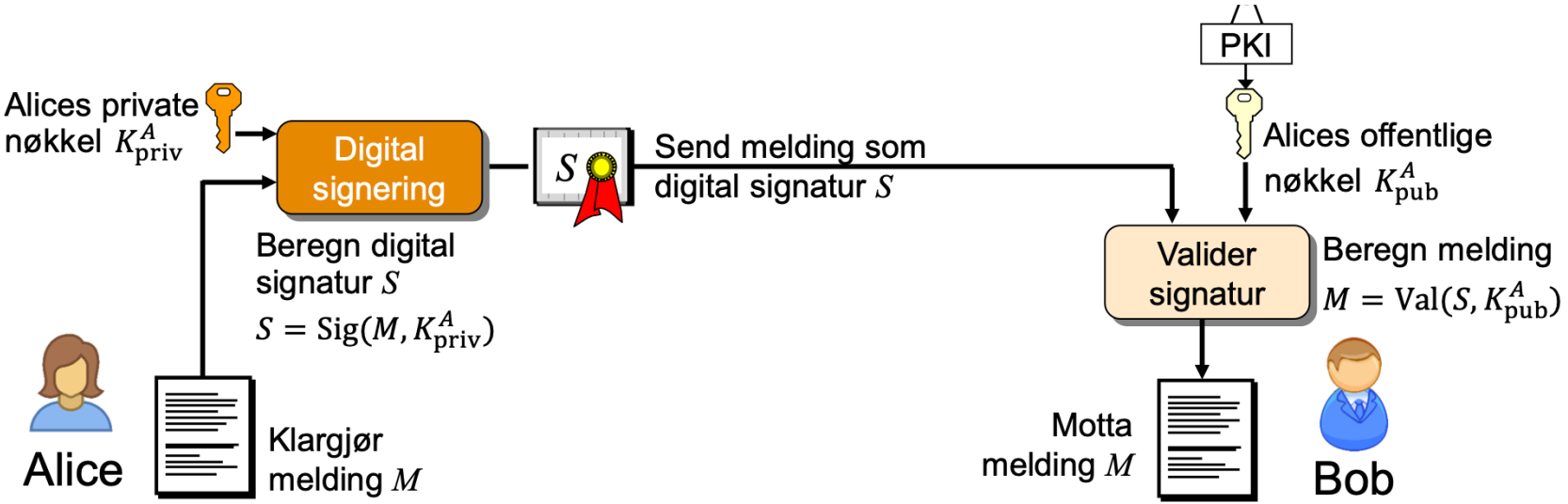
# Asymmetrisk kryptering




*Illustrasjon hentet fra IN2120 Informasjonssikkerhet*

# Digital signatur

Det motsatte av det vi så med vanlig asymmetrisk kryptering




Gir uavviselighet også

*Illustrasjon hentet fra IN2120 Informasjonssikkerhet*

# Praktiske oppgaver (løses på ifis maskiner)

- Nettverk:
  - DNS-oppslag
  - Arp-cache
  - Traceroute
  - Chattetjeneste
- Sikkerhet:
  - Sjekksumalgoritme (hashing)

Eventuelt gjennom ssh: ssh brukernavn@login.ifi.uio.no

# DNS-oppslag

- Programmene på din datamaskin kan bare sende beskjeder til andre maskiner hvis den kjenner IP adressen til maskinen den skal snakke med. Vi foretrekker å bruke navn i stedet for en IP-adresse. Maskinen huldra.uio.no har for eksempel adressen 129.240.2.27
- Åpne opp en terminal i Linux og bruk kommandoen “dig” for å finne adresse for navnene.
- Hva finner du ut hvis du for eksempel kjører "dig ifi.uio.no" og dig "login.ifi.uio.no". Hva med "dig www.microsoft.com".

# Hva er ARP-cache?

- Åpne opp en terminal i Linux og bruk kommandoen «arp»
- Hvorfor trenger vi ARP-cache?

# Traceroute

- Når du programmerer på applikasjonslaget bruker vanligvis programmet transportlaget, som igjen bruker lagene under. Kommandoen "traceroute" brukes for å vise stien som pakken bruker fra din datamaskin til destinasjonen.
- Hvor mange routere er brukt for å sende en pakke til www.ifi.uio.no, www.uio.no, www.ntnu.no, www.kth.se, www.cmu.edu
- Hvis ikke programmet "traceroute" er installert på din maskin kan du bruke et online verktøy: http://networktools.nl/traceroute/

- Finn en maskin på hvert kontinent, og kjør traceroute til dem. Tips: For å finne en maskin på et bestemt sted, kan du forsøke å søke etter et universitet eller en bedrift som hører til på stedet og prøve traceroute til hjemmesidens domenenavn.
- Bruk verktøyet Traceroute Mapper til å analysere rutene dere har funnet
- https://stefansundin.github.io/traceroute-mapper/
- Bonusoppgave: Kjør kommandoen "traceroute bad.horse"

# Chattetjeneste

- Til denne chattetjenesten skal vi bruke kommandoen nc. (netcat) Netcat lar oss sende pakker mellom ulike maskiner, og det er det vi skal gjøre i denne oppgaven!
- På den ene verten (et terminalvindu) skriv inn kommandoen: nc -l <portnummer>
  - Det denne kommandoen gjør er å be verten om å lytte på det portnummeret som er oppgitt.
- Videre må vi vite hvilken IP-adresse denne verten har for å kunne sende meldinger til den!
  - For dette bruker vi kommandoen ifconfig
- Åpne en ny terminal/en annen maskin. Skriv inn kommandoen: nc <ip-adresse-fra-ifconfig> <portnummer>
  - På den måten kobler vi oss til denne lyttende porten
- Voila! Send meldinger til hverandre ☺

# Sjekksumalgoritme (hash-funksjon)

1. a) Programmet sha256sum genererer en sjekksum av en fil basert på algoritmen SHA256. Lag en liten tekstfil med innhold, kall den f.eks. abc.txt. Kjør så programmet sha256sum: [kritisk@vestur]>sha256sum abc.txt [6ee0c32c675ce6d3bd3f6e326c81e45b3d6675c29c0c9ced1398684a667804e9 abc.txt
2. Gjør endringer i fila abc.txt, og kjør programmet en gang til. Er nøkkelen den samme?
3. b) Hvordan kan sjekksumalgoritmer bidra til å sikre dataintegritet?
4. c) Finn og diskuter situasjoner/eksempler hvor bruk av sjekksumalgoritme kan være nyttig for deg.