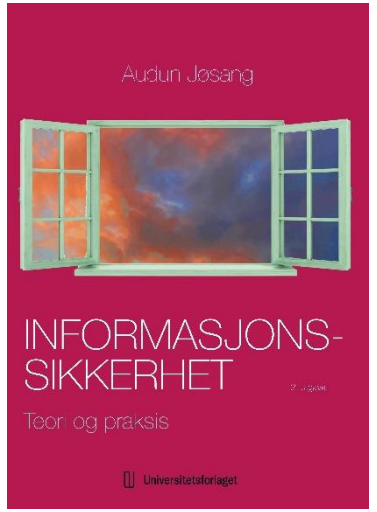
Kapittel 5:
Nøkkelhåndtering og PKI

Informasjonssikkerhet: Teori og praksis

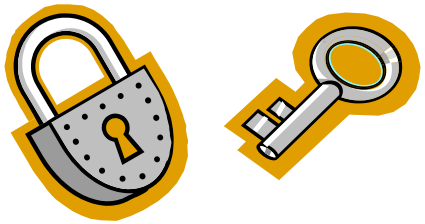
Audun Jøsang

2. utg. 2023

Universitetsforlaget



Oversikt

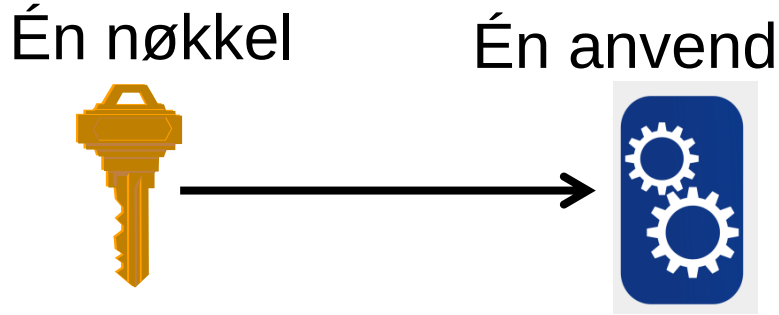


- Nøkkelhåndtering
  - Hvorfor nøkkelhåndtering
  - Kryptoperioder
  - Anbefalte nøkkellengder
  - Prosess for nøkkelhåndtering
- PKI
  - Utfordringen med nøkkeldistribusjon
  - Viktighet av autentisitet for offentlige nøkler
  - X-509-sertifikater, generering og validering
  - Tillitsstrukturer
  - Internett-PKI

Viktigheten av nøkkelhåndtering for kryptografi

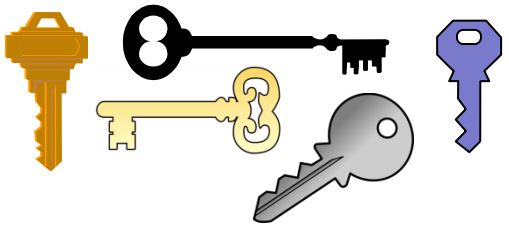
- Styrken til kryptografiske systemer avhenger av:
  - Størrelsen på nøkkelen
  - Styrken til kryptografiske algoritmer/protokoller
  - Korrektheten og integriteten til HW/SW -implementeringen av kryptografiske algoritmer/protokoller
  - Nøkkelhåndtering
- Nøkkelhåndtering gir grunnlaget for sikker generering, lagring, distribusjon og destruering av nøkler.
- Dårlig nøkkelhåndtering kan lett føre til brudd på sikkerheten av systemer som er basert på kryptografi.

Én anvendelse per nøkkel



- En enkelt nøkkel skal bare brukes til en enkelt anvendelse
  - f.eks. kryptering, autentisering, nøkkelinnkapsling, generering av tilfeldige tall eller for digital signatur
- Bruk av den samme nøkkelen til flere anvendelser kan svekke sikkerheten til en eller flere anvendelser.
- Begrensning av bruk av en nøkkel begrenser skaden som kan oppstå hvis nøkkelen blir kompromittert.
- Samme prinsipp om at passord ikke skal gjenbrukes mellom forskjellige tjenester.
  - Gjenbruk av passord kan gjøre mange tjenester usikre hvis passordet lekker eller blir cracket på en av tjenestene.

Typer av kryptografiske nøkler



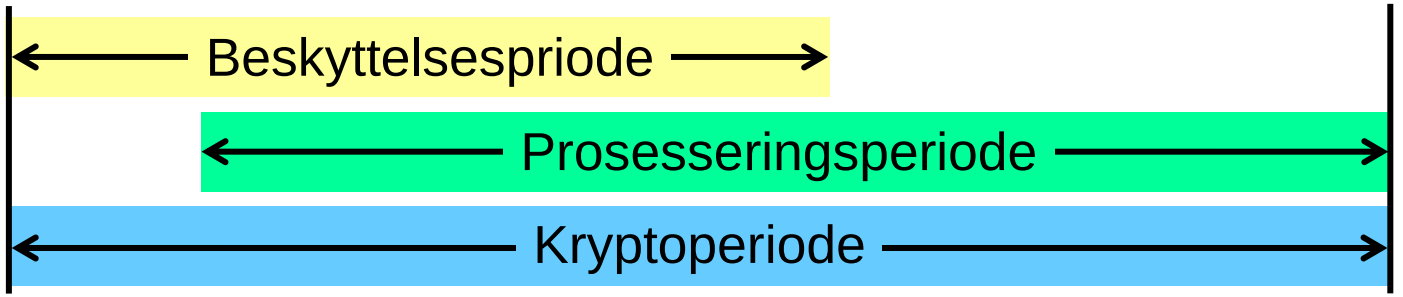
- Kryptonøkler er klassifisert i henhold til:
  - om de er offentlige, private eller symmetriske
  - tiltenkt bruk
  - om de er statisk (langt liv) eller flyktig (kort levetid)
- 19 forskjellige typer kryptografiske nøkler definert i : NIST Special Publication 800-57, Part 1, “Recommendation for Key Management”

  https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-57pt1r5.pdf

Kryptoperiode

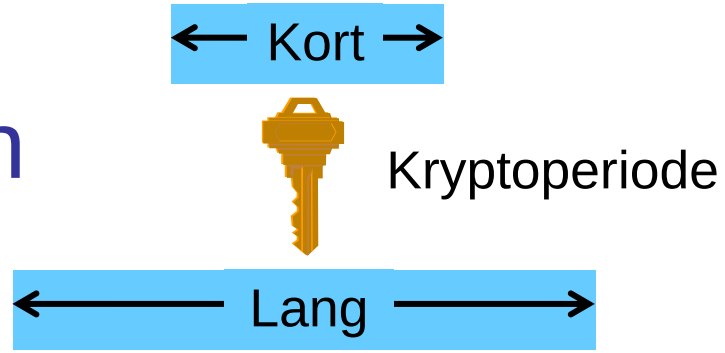
- Kryptoperioden er tidsperioden for godkjent bruk av en bestemt nøkkel
- Kryptoperioden er viktig fordi den:
  - Begrenser mengden data som er beskyttet av en gitt nøkkel, som potensielt kan kryptoanalyseres.
  - Begrenser mengden eksponering og skade, hvis en enkelt nøkkel blir kompromittert.
  - Begrenser bruken av en bestemt algoritme til den estimerte sikre levetiden.

Kryptoperioder



- En nøkkel kan brukes til beskyttelse og/eller prosessering.
- Beskyttelsesperiode:
  - Nøkkelen brukes til å kryptere (hemmelig symmetrisk eller offentlig asymmetrisk nøkkel)
  - Nøkkelen brukes til å generere en digital signatur (privat asymmetrisk nøkkel)
- Prosesseringsperiode:
  - Nøkkelen brukes til å dekryptere (hemmelig symmetrisk eller privat asymmetrisk nøkkel)
  - Nøkkelen brukes til å validere en digital signatur (offentlig asymmetrisk nøkkel)
- Kryptoperioden varer fra begynnelsen av beskyttelsesperioden til slutten av prosesseringsperioden
  - Prosesseringsperioden kan fortsette etter beskyttelsesperioden

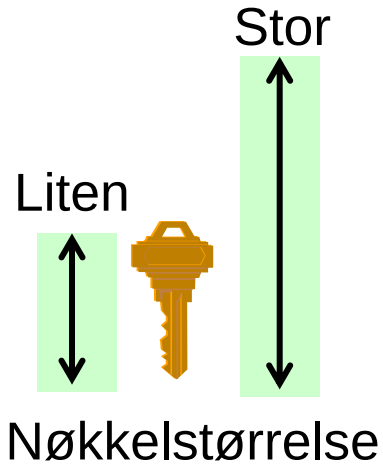
Faktorer som bestemmer kryptoperioden



- Generelt, ettersom sensitiviteten til informasjonen eller tjenesten øker, bør kryptoperioden reduseres for å begrense potensiell skade i tilfelle kompromitering.
- Korte kryptoperioder kan være kontraproduktive, spesielt der hvis tilgjengelighet av data/tjeneste er viktig, og det er betydelig overhead og/eller en viss sannsynlighet feil i nøkkelgenerering eller nøkkeldistribusjonsprosessen.

- Lengden på kryptoperioden er derfor en avveining

Anbefalte størrelser (i bits) på symmetriske (hemmelige) nøkler



- Symmetriske hemmelige nøkler brukes i blokkchiffer som AES.
- Størrelsen på nøkkelen bestemmer gjennomsnittlig tidsforbruk for kryptoanalyse med utmattende søk, dvs. å prøve alle nøkler.
- Stadig kortere tid for utmattende søk pga. stadig større regnekraft.
- Eiere av systemer må sette nøkkelstørrelse ut ifra krav om hvor mange år et kryptosyster skal forbli sikkert.

Anbefalte størrelser for symmetriske nøkler

Ref: NIST SP 800-57

Security Strength		Through 2030	2031 and Beyond
< 112	Applying protection	Disallowed	
	Processing	Legacy use	
112	Applying protection	Acceptable	Disallowed
	Processing		Legacy use
128	Applying protection and processing information that is already protected	Acceptable	Acceptable
192		Acceptable	Acceptable
256		Acceptable	Acceptable

Ekvivalente størrelser for asymmetriske nøkler

Ref: NIST SP 800-57

		Finite Field Cryptography	Integer Factorization Cryptography	Elliptic Curve Cryptography
Security Strength	**Symmetric Key Algorithms**	**FFC (DSA, DH, MQV)**	**IFC* (RSA)**	**ECC* (ECDSA, EdDSA, DH, MQV)**
128	AES-128	$L = 3072$ $N = 256$	$k = 3072$	$f = 256\text{-}383$
192	AES-192	$L = 7680$ $N = 384$	$k = 7680$	$f = 384\text{-}511$
256	AES-256	$L = 15360$ $N = 512$	$k = 15360$	$f = 512+$

* The security-strength estimates will be significantly affected when quantum computing becomes a practical consideration.

Kvantecomputeres potensielle trussel mot kryptografi

- NIST (US National Institute of Standards and Technology) antyder muligheten for kraftige kvantecoputere på slutten 2020 –tallet

- Konsekvens for tradisjonell asymmetrisk krypto:
  - RSA
  - Elliptisk kurvekryptografi (ECDSA)
  - Finite Field Cryptography (DSA)
  - Diffie-Hellman nøkkelutveksling

➢ Behov for nye postkvantealgoritmer

- Konsekvens for symmetrisk krypto:
  - AES

➢ Dobbel nøkkelstørrelse (256 bits) er OK

- Konsekvens for hashfunksjoner:
  - SHA-2 og SHA-3 med 512-bits hash

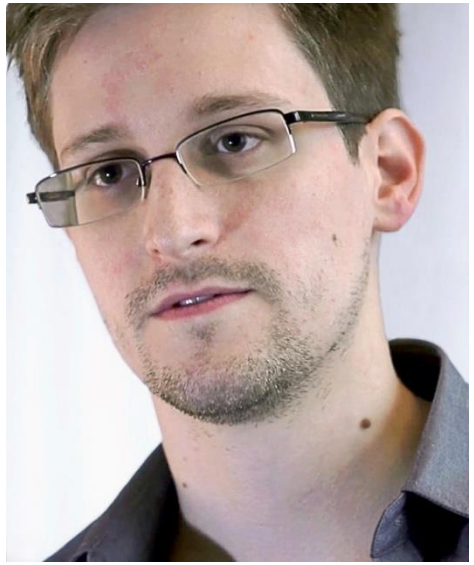
➢ Dobbel hashstørrelse (512 bits) er OK

Nøkkelgenerering

- Mest sensitiv av alle kryptografiske operasjoner.
- Nøkkelgeneratorer i programvare eller maskinvare må beskyttes for å forhindre:
  - lekkasje, svekking eller forfalskning av nøkler,
  - lekkasje, svekking eller forfalskning av IV (initialiseringsvektorer).
- Nøkler må velges tilfeldig fra hele nøkkelrommet
  - f.eks. 128 bits nøkkel gir et nøkkelrom på 2^{128} forskjellige nøkler
  - Hver nøkkel bør være like sannsynlig

Når nøkler ikke er tilfeldige

- Edward Snowden avslørte i 2013 at RSA Security Inc. (fremtredende sikkerhetsselskap) ble bestukket av NSA med 10 millioner dollar for å implementere en svak generator for tilfeldige tall i sine BSAFE-sikkerhetsprodukter.
- NSA var i stand til å forutsi tilfeldige tall og regenerere de samme hemmelige nøklene som de som ble brukt av RSAs kunder.
- Med de hemmelige nøklene kunne NSA lese alle data kryptert med RSAs BSAFE sikkerhetsprodukt.



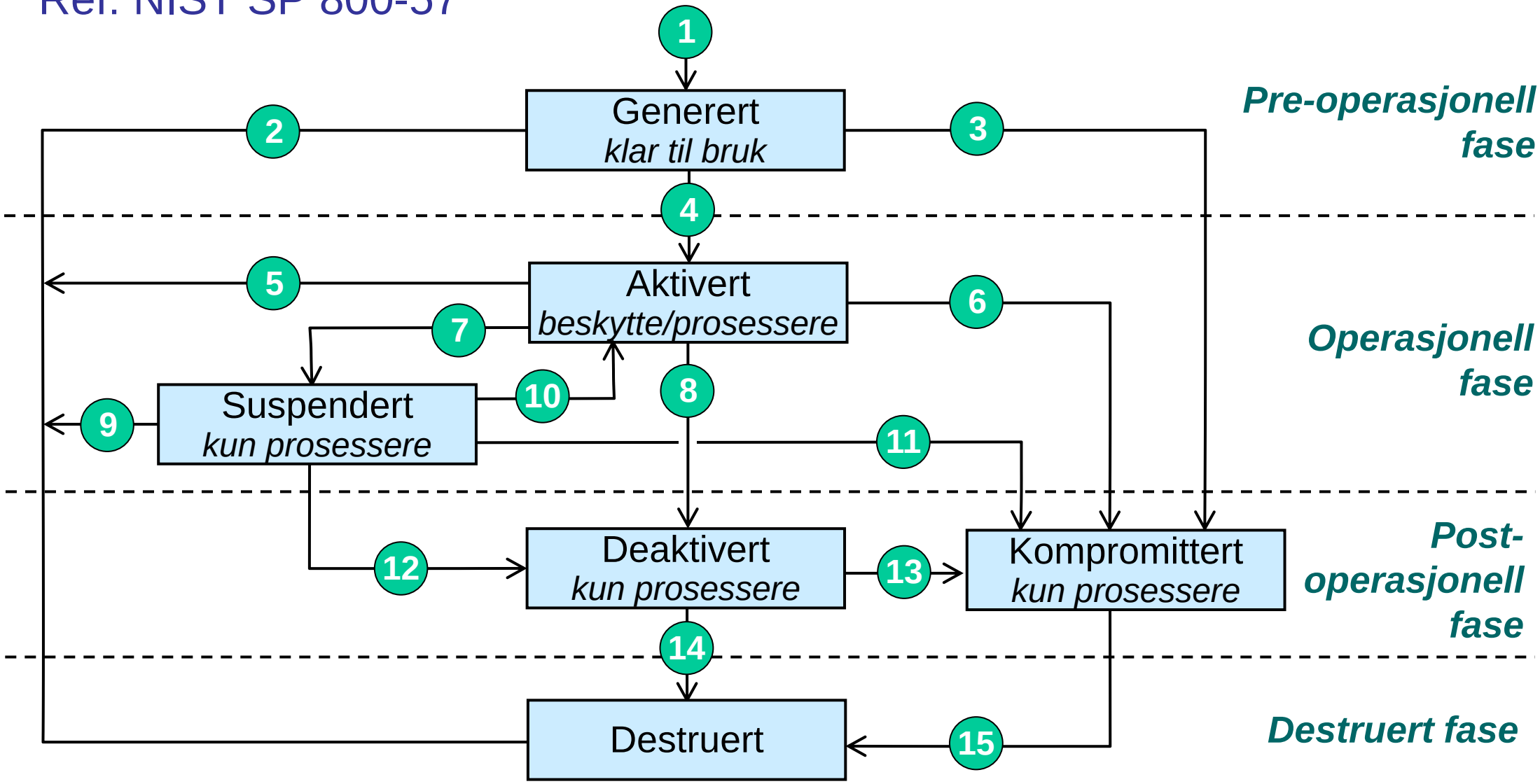
Edward Snowden

Kompromittering av kryptografiske nøkler

- Kompromittering av nøkler er hvis det er kjent eller mistenkt at en uautorisert enhet har skaffet seg en hemmelig/privat nøkkel.
- Når en hemmelig nøkkel er kompromittert, må bruk av nøkkelen for beskyttelse opphøre umiddelbart, og nøkkelen defineres som kompromitert.
- En kompromittert nøkkel kan brukes til fortsatt prosessering (dekryptering) av kryptert data.
- Alle brukere av nøkkelen må informeres og gjøres klar over risikoen.
- Fortsatt nøkkelbruk for behandling avhenger av risikoen og av organisasjonens retningslinjer for nøkkelhåndtering.
- Nøkkelkompromittering er alvorligst er når det skjer uten av det oppdagelse.

Nøkkeltilstander, transisjoner og faser

Ref: NIST SP 800-57



Beskyttelse av nøkler

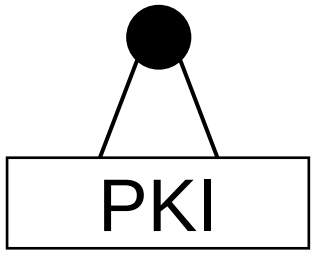
- Hemmelige nøkler for symmetriske chiffer
  - Aldri lagret eller overført "klart"
  - Kan bruke hierarki: sesjonsnøkler kryptert med masternøkkel
  - Beskyttelse av masternøkkel:
    - Låser og vakter
    - Manipuleringssikre enheter
    - Beskyttelse med passord
- Nøkler for asymmetriske chiffer
  - Private nøkler trenger konfidensialitetsbeskyttelse (som for hemmelige nøkler)
  - Offentlige nøkler trenger beskyttelse av integritet/autentisitet (se neste avsnitt “PKI”)

Nøkkeldestruering

- Nøkler må ikke eksistere i flyktig minne eller på permanente lagringsmedier etter destruering
- Meetoder for destruering av nøkler er f.eks.
  - Enkel sletteoperasjon på system
    - Fare for at nøkkelen fremdeles fins i papirkurven eller på disksektorer
  - Spesiell sletteoperasjon på datamaskinen
    - som f.eks. overskriver minne og disk slik at det ikke etterlater restdata,
  - Magnetisk «degaussing» av magnetiske lagringsmedier med sterkt magnetfelt
  - Ødeleggelse av fysisk enhet f.eks. med knusing eller høy temperatur
  - Destruering av masternøkkel vil logisk også destruere underordnede nøkler kryptert under masternøkkelen

PKI (Public-Key Infrastructure) Offentlig nøkkelinfrastruktur

Kryptografi løser sikkerhetsutfordringer i åpne datanett,
men skaper utfordringer for nøkkeldistribusjon.



Asymmetrisk kryptografi forenkler nøkkeldistribusjonen, men krever
PKI som skaper utfordringer for tillitshåndtering.

Utfordring for nøkkeldistribusjon

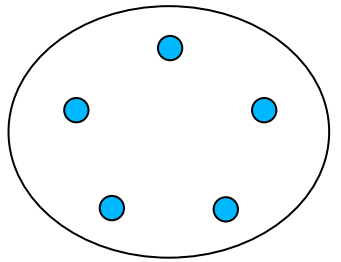
- Anta et datanett (f.eks. Internett) med *n* noder
- Hvert nodepar trenger en separate nøkkel for å kunne kommunisere sikkert med kryptografisk beskyttelse
- Hvor mange sikre nøkkeldistribusjoner er nødvendig?
  - Symmetriske hemmelige nøkler, krever **konfidensialitet**.

    $n(n\text{-}1)/2$ distribusjoner, vokser kvadratisk.

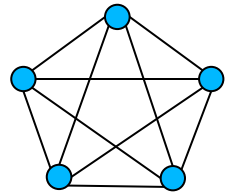
    upraktisk i åpne nettverk.
  - Asymmetriske offentlige nøkler, krever **autentisitet.**

    $n(n\text{-}1)$ distribusjoner av offentlige nøkler, vokser kvadratisk

    upraktisk i åpne nettverk
  - Asymmetriske offentlige nøkler med PKI, krever **autentisitet**.
    - 1 rotnøkkel distribueres til alle *n* noder
    - vokser lineært
    - … mye lettere, men likevel relativt utfordrende

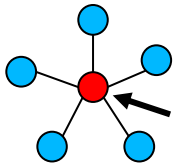


datanett



n noder

$n(n\text{-}1)/2$ kanter



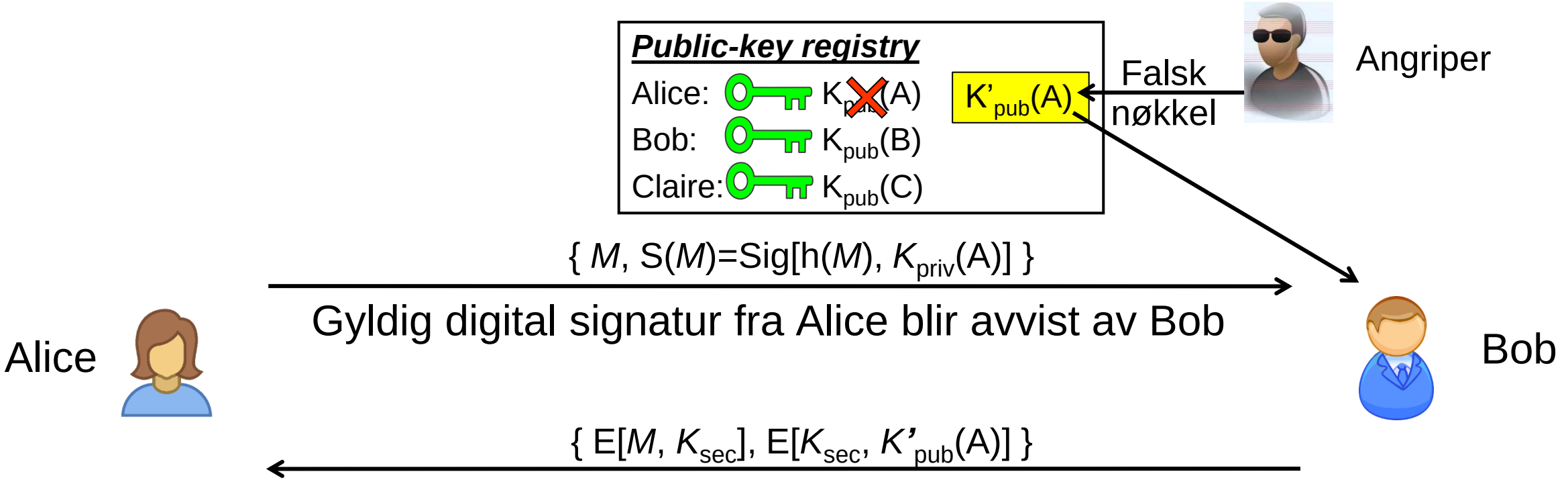
rot

n noder

n kanter

Problemet med forfalskning av offentlige nøkler

- Anta at offentlige nøkler lagres i et offentlig register
- Hva er konsekvensen hvis angriper forfalsker Alices offentlige nøkkel i registret?

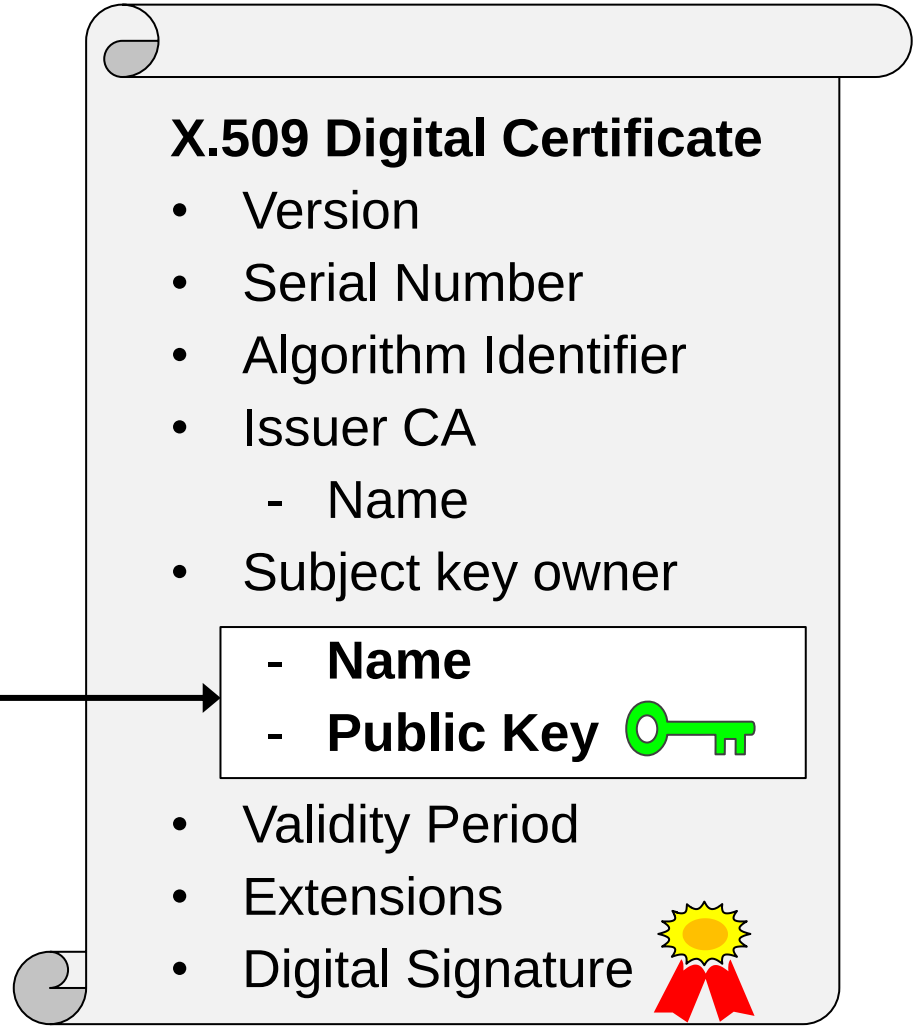


PKI: Infrastruktur for sikker distribusjon av offentlige nøkler

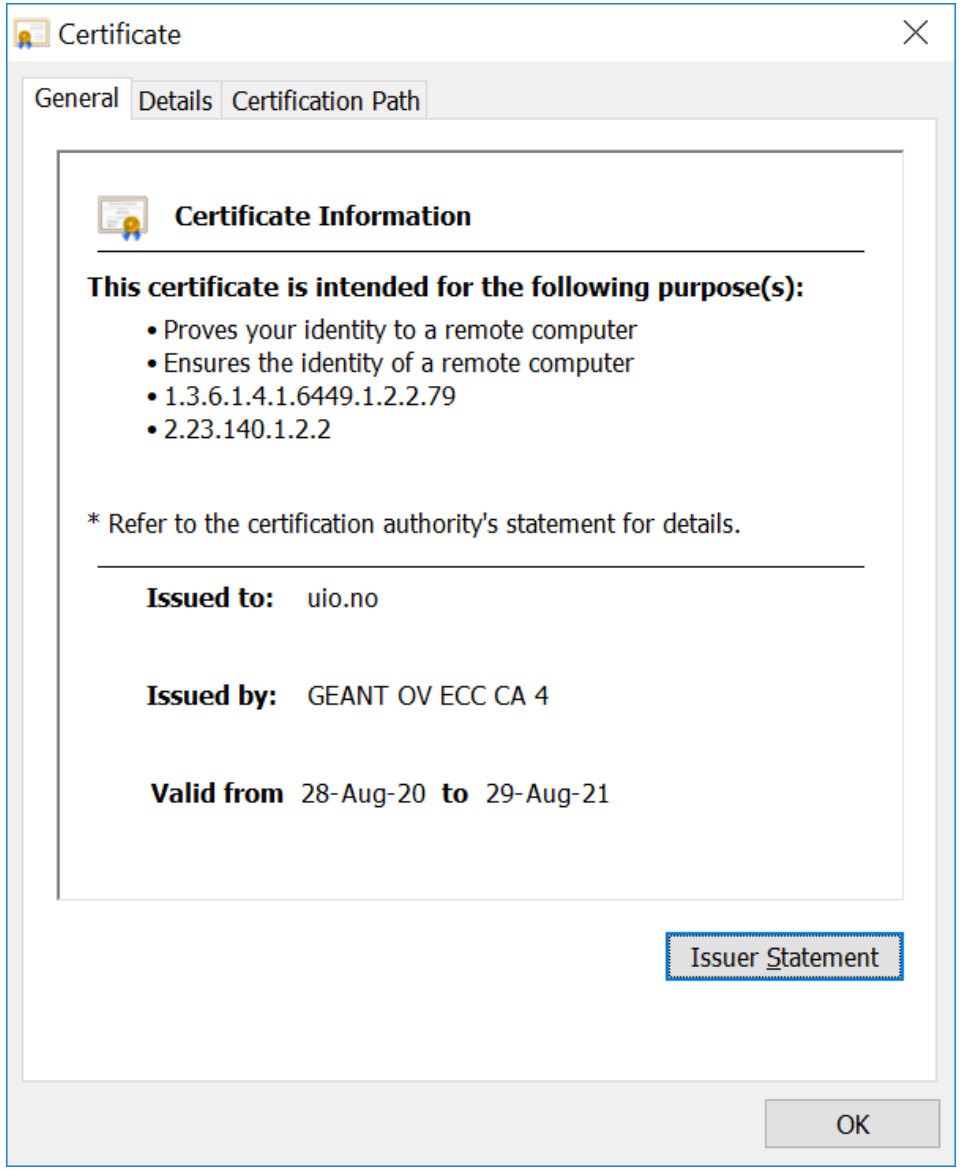
- For å beskytte autentisitet må offentlige nøkler signeres digitalt og distribueres som "offentlige nøkkelsertifikater".
- Formålet med en PKI er garantere autentisitet av offentlige nøkler og forenkle nøkkeldistribusjonen. PKI består bl.a. av:
  - **Policyer** (for å definere reglene for forvaltning av sertifikater)
  - **Teknologier** (for å generere, distribuere, lagre og validere sertifikater)
  - **Prosedyrer** (knyttet til forvaltning av sertifikater)
  - **Tillitsmodell** for offentlige nøkkelsertifikater (hvordan sertifikatene er kryptografisk knyttet til hverandre)

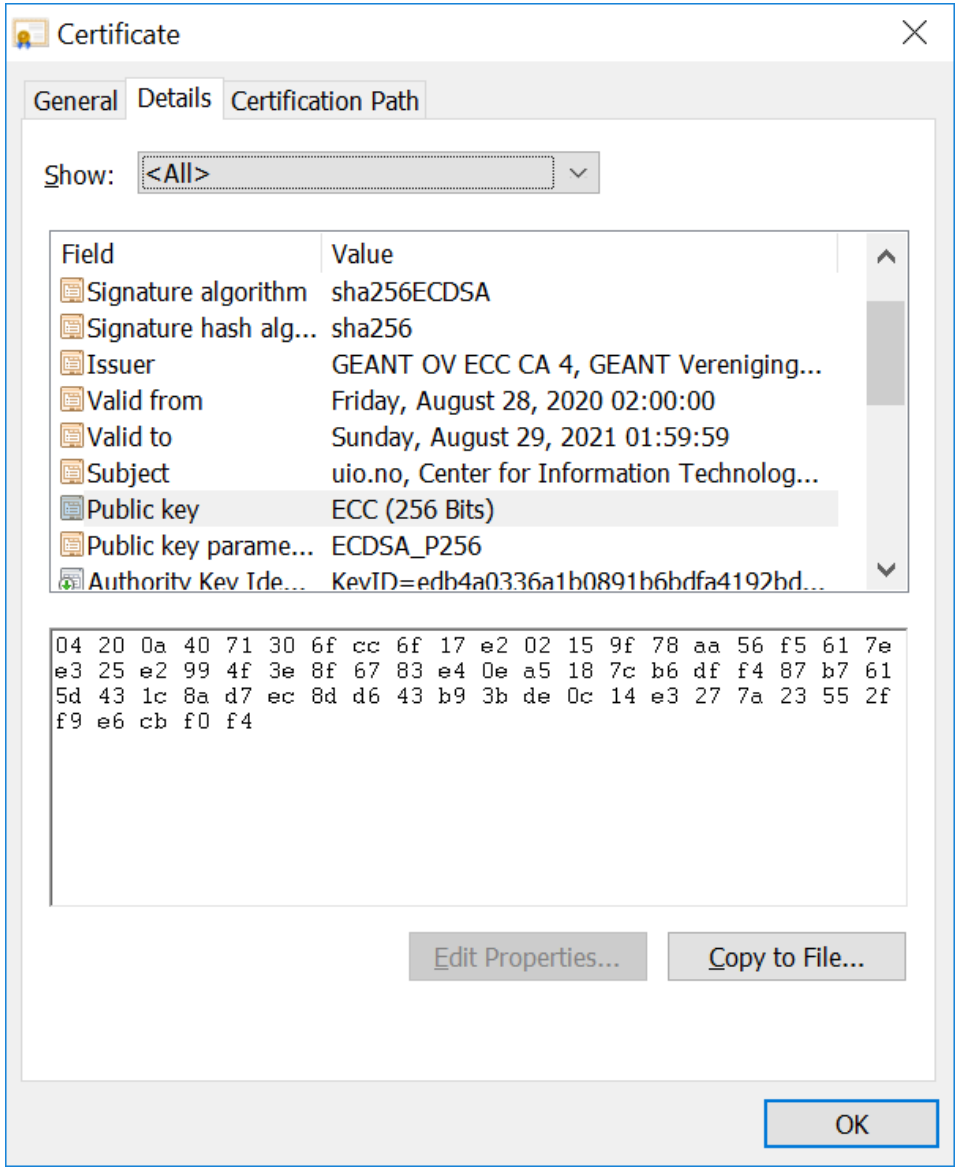
X.509 offentlig nøkkelsertifikater

- Format for public-key-sertifikater er definert i X.509-standarden
- Et sertifikat med offentlig nøkkel er en record med data, inkludert subjektets (domene)navn og dets offentlige nøkkel, alt digitalt signert av en CA (Certificate Authority).
- Skaper logisk binding mellom (domene)navnet og dets offentlige nøkkel
- En autentisk kopi av CAs offentlige nøkkel er nødvendig for å validere sertifikatet dvs. verifisere at den digitale signaturen er korrekt.

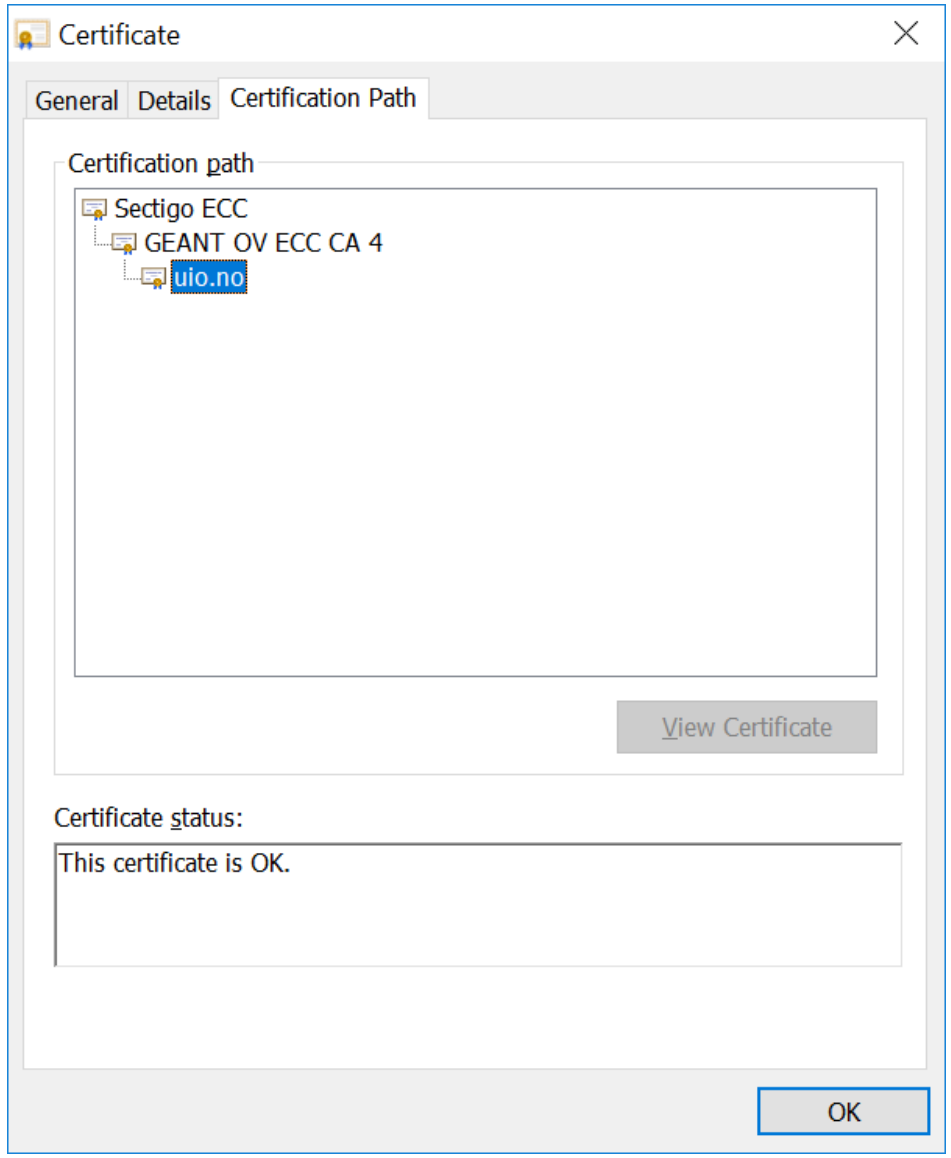


Eksempel på X.509 sertifikat





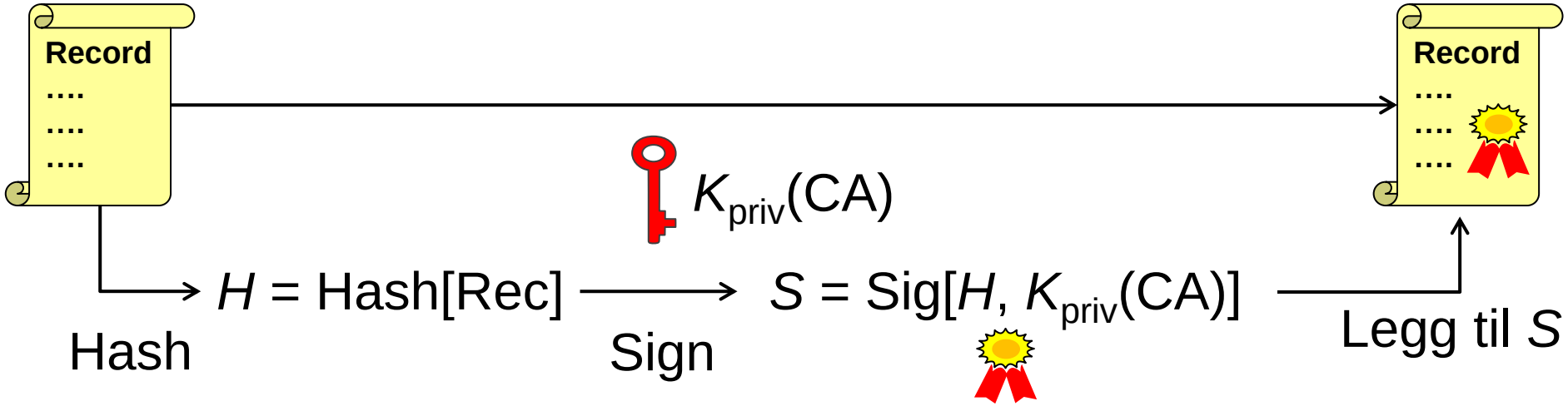
Eksempel på X.509 sertifikat (forts.)



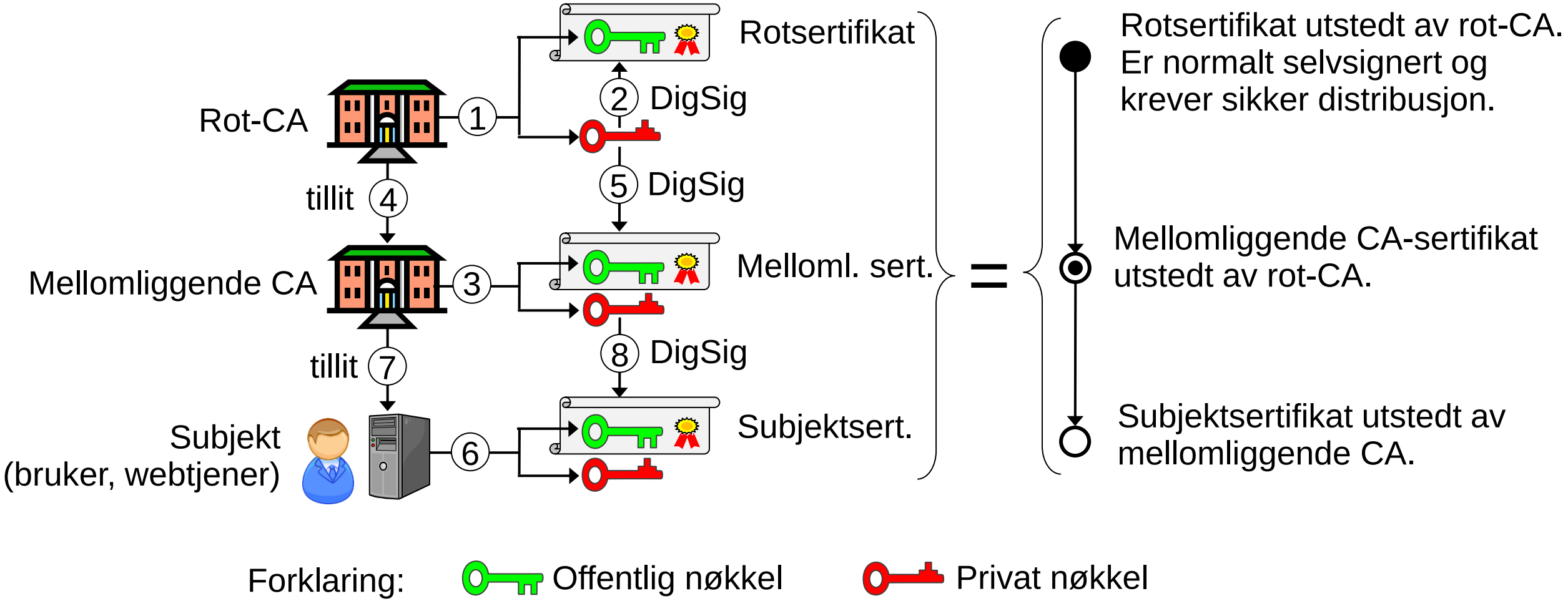
- Sertifikatstien er kjeden med X.509 - sertifikater fra roten til subjektets sertifikat
- Rotsertifikatet er forhåndslagret som en del av nettleseren
- Hver nettleser lagrer noen hundre rotsertifikater.
- Mellomliggende sertifikater kan forhåndslagres eller lastes ned i sanntid.
- Validering av subjektsertifikater starter fra roten.

Hvordan generere et offentlig nøkkelsertifikat

1. Samle alle datafeltene, inkludert subjektets (domene)navn og offentlige nøkkel, i en datarecord Rec
2. Hash datarecorden
3. Signer hash-verdien
4. Legg den digitale signaturen til datarecorden



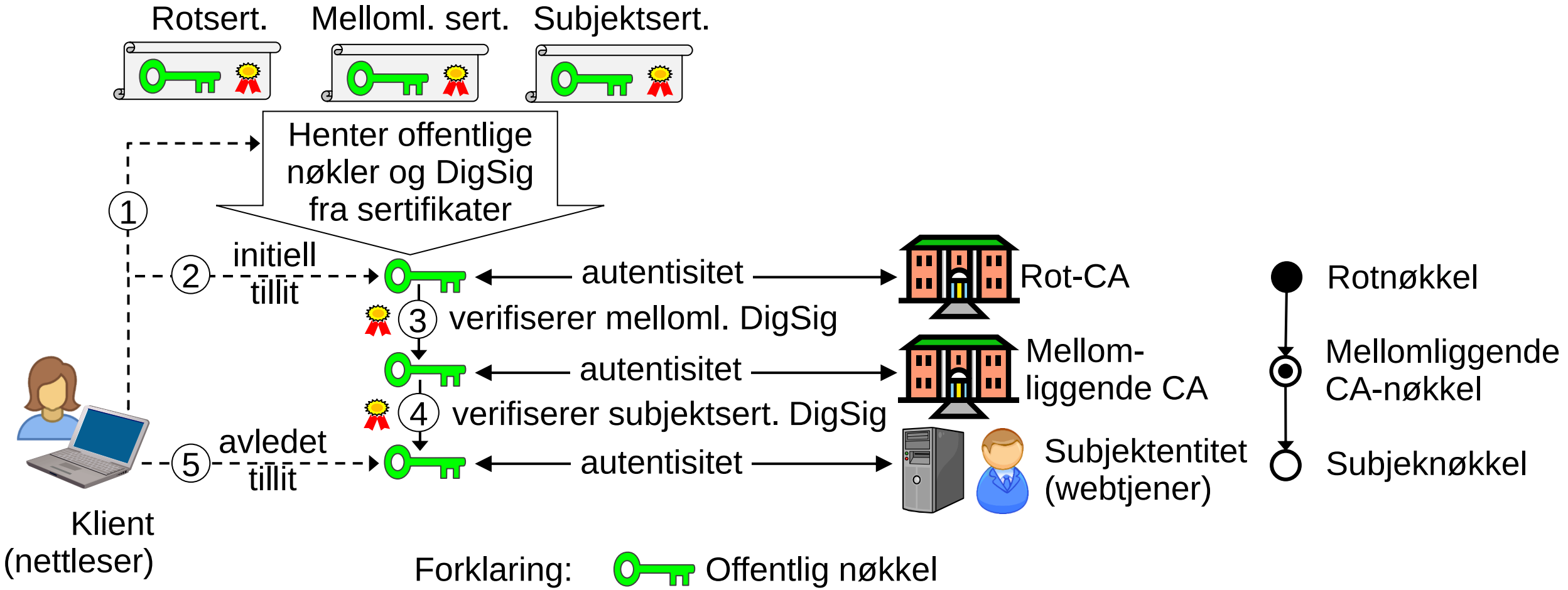
Kjede av sertifikater



Hvorfor selv-signerte rotsertifikater?

- Noen tror feilaktig at et rotsertifikat er autentisk bare fordi det er selvsignert.
- Begrepet "selvsignert" kan være villedende
  - Kan gi et falskt inntrykk av sikkerhet
  - Kan brukes til å skjule falske sertifikater
  - Kan brukes til forfalsking av serversertifikater i TLS -strippeangrep
- Selvsignering gir absolutt ingen sikkerhet
- Mulige nyttige formål med selvsignering:
  - Gir en sjekksum for å oppdage utilsiktet korrupsjon
  - X.509 -sertifikater har et felt for digital signatur, så et tomt felt kan føre til at programmer ikke fungerer. En selv-signatur er en måte å fylle det tomme feltet på

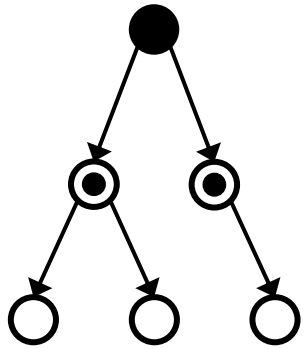
Validering av sertifikater



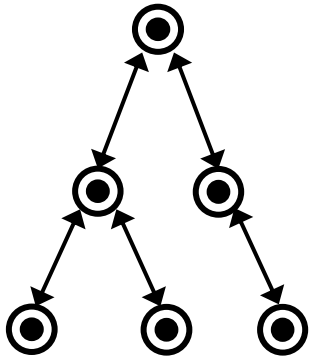
Tillits-modeller

Forklaring:

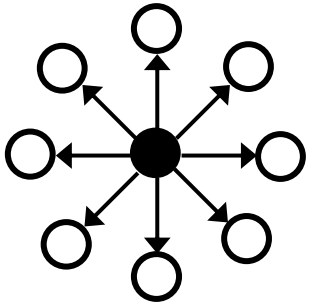
- Selvsignert rot-CA-sertifikat
- CA-signert mellomliggende CA-sert
- CA-signert subjektsertifikat
- Selvsignert subjektsertifikat



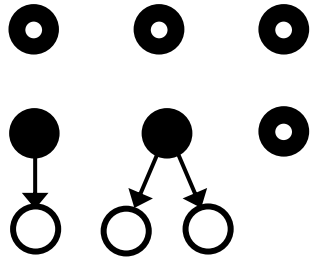
Enkelt hierarki



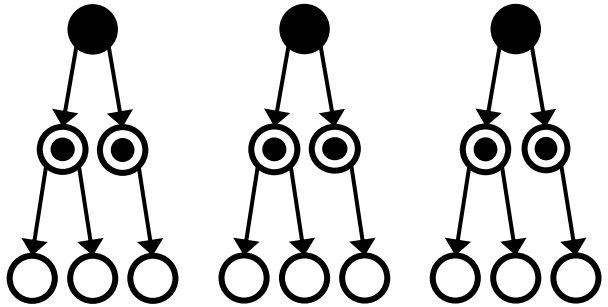
Bi-direksjonelt hierarki



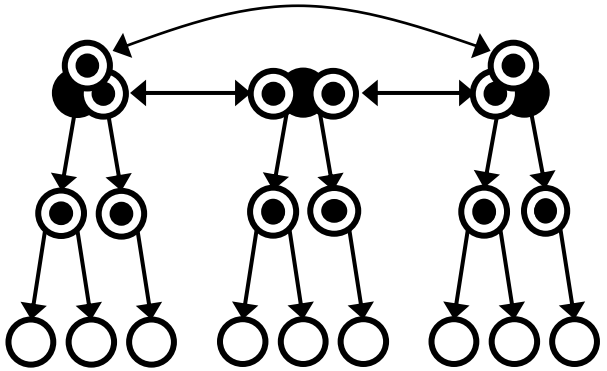
Bruker-sentrisk
(bruker er sin egen CA)



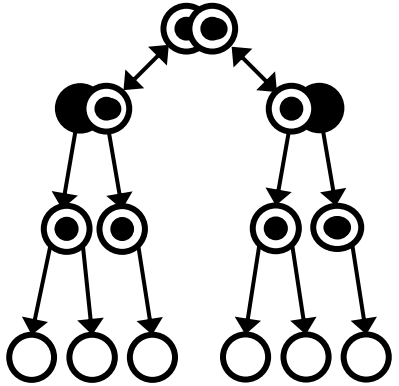
Ustrukturert PKI



Silo-hierarkier
(Internett-PKI)

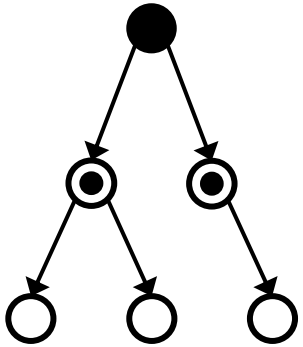


Kryss-sertifiserte hierarkier
(Mesh-PKI)



PKI-er med bro-CA

Enkelt hierarki



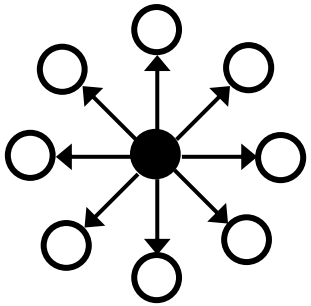
Forklaring:

- Selvsignert rot-CA-sertifikat
- CA-signert mellomliggende CA-sertifikat
- CA-signert subjektsertifikat

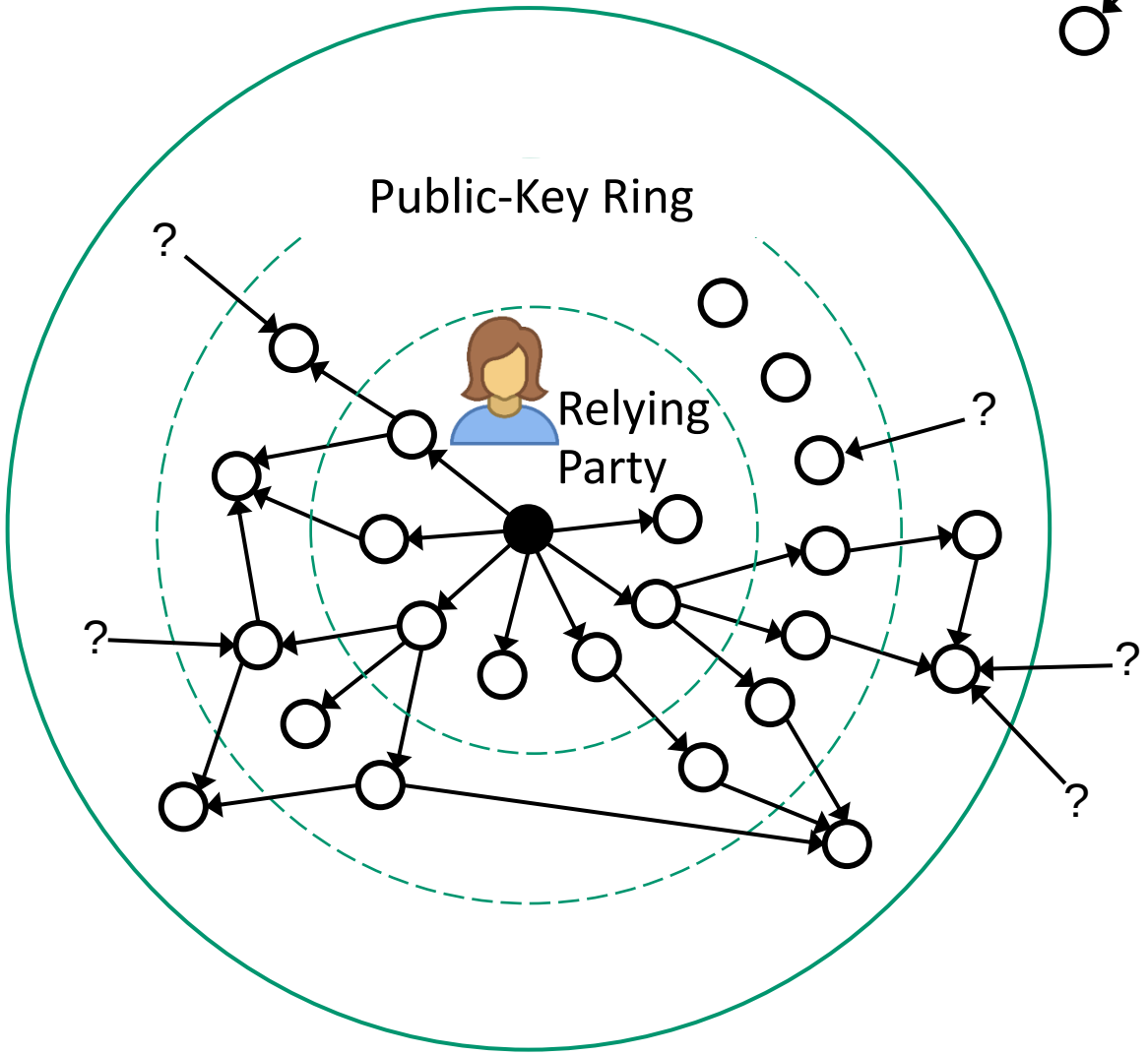
- Fordeler:
  - fungerer godt i høyt strukturerte organisasjoner som militære nett
  - Enkel tillitsstruktur
  - Fungerer godt i lukket/isolert datanett
- Ulemper:
  - Alle subjekt-entiteter må ha tillit til samme rot-CA
  - Kompromittering av rot-CA fører til totalt sammenbrudd av sikkerhet
  - Skalerer ikke til åpne datanett

Bruker-sentrisk PKI

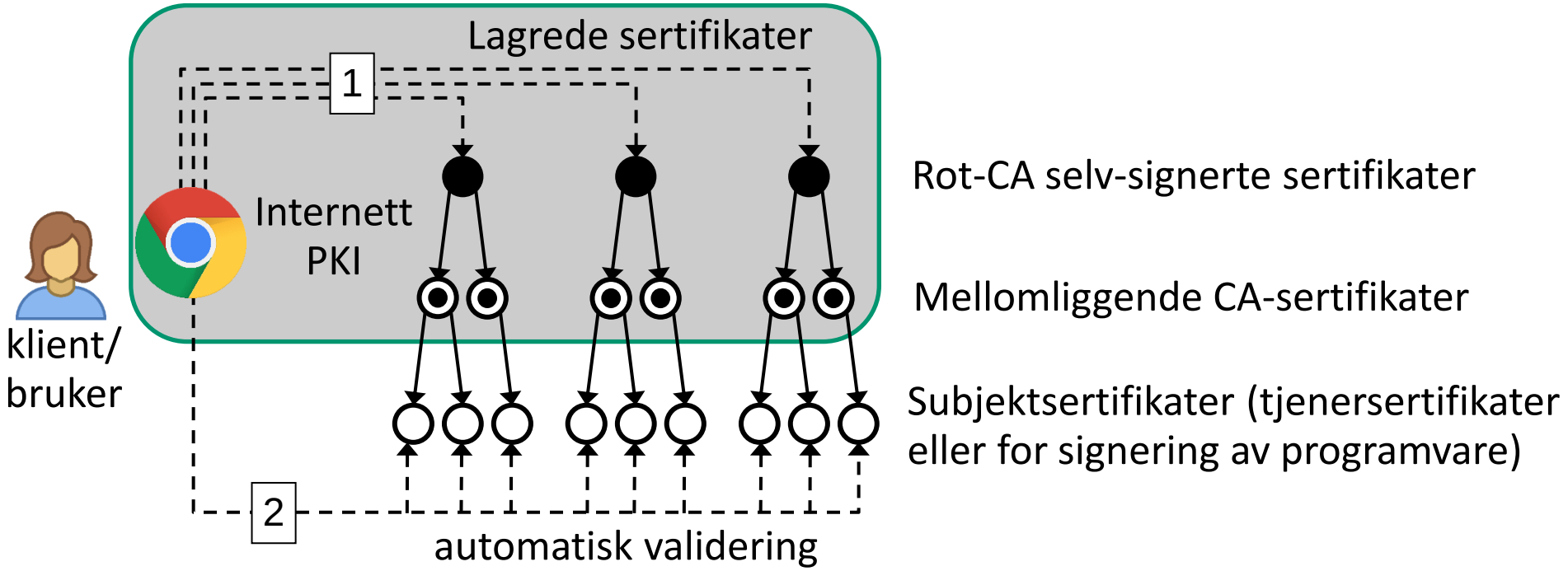
Hver bruker er sin egen rot-CA



- Hver entitet signerer offentlige nøkler til andre når de er bekreftet å være autentiske.
- Hver bruker har sin egen "offentlige nøkkelring"
- Offentlige nøkler signert av andre pålitelige personer kan også betraktes som autentiske.
- Denne modellen brukes i PGP (også kalt GPG)



Internett PKI: Silo-hierarkier
(PKI som brukes av nettlesere for https)

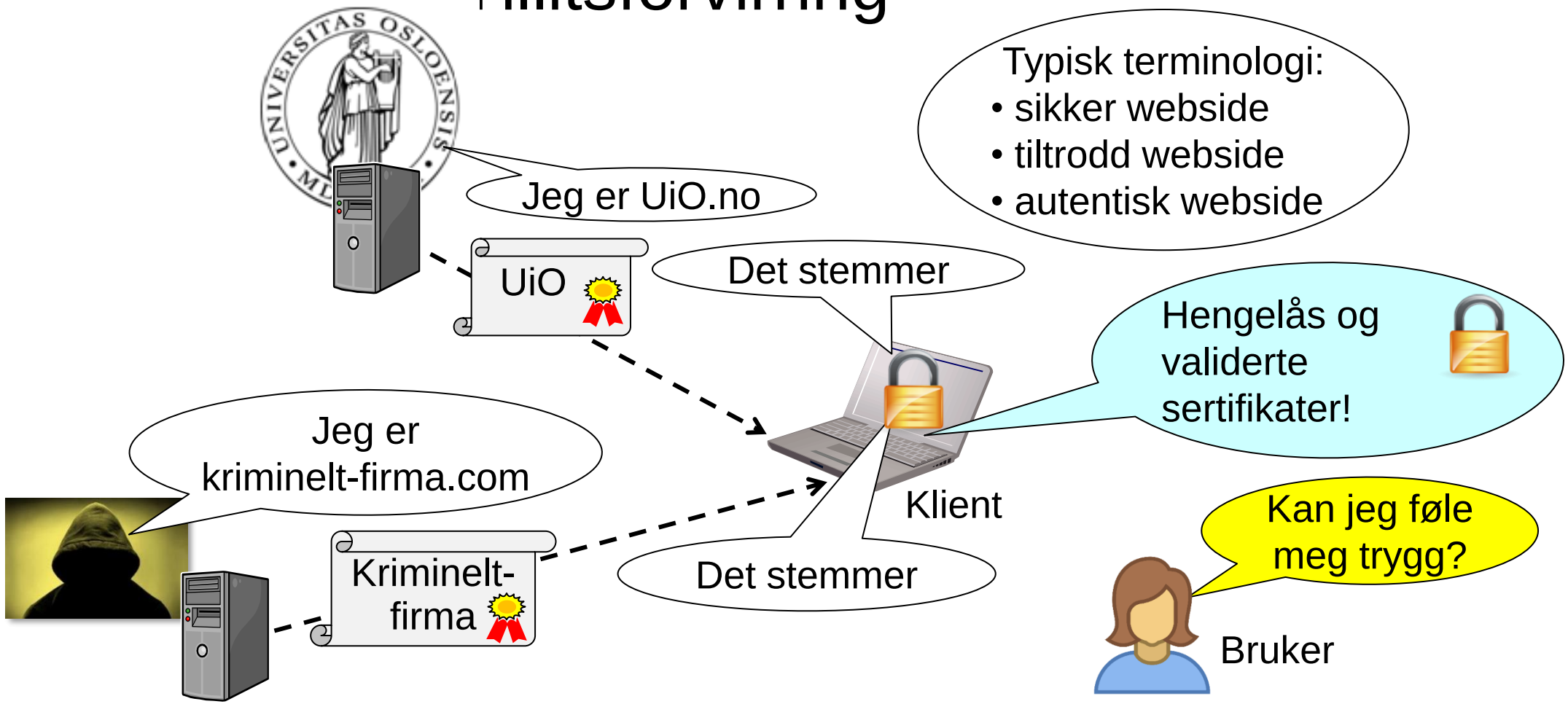


Internett-PKI består av noen hundre silo-hierarkier, der (rot) CA-sertifikatet for hvert hierarki er installert som en del av nettleseren. Hvert enkelt PKI er sin egen silo, uten forbindelse til andre enkelte PKI-er. Nye/oppdaterte rot-sertifikater kan importeres av bruker eller gjennom (automatisk) oppdatering av programvare

Internett-PKI og falske sertifikater

- Sertifikater valideres automatisk ved at nettleseren sjekker den digitale signaturen, og det er samsvar mellom sertifikatets domenenavn nettsidens domenenavn
- Kriminelle kan kjøpe legitime sertifikater som automatisk valideres av nettlesere
- Legitime sertifikater kan brukes sammen med phishing-angrep, f.eks. for å lage en falsk nettside for en bank
- Falske nettsider kan ha legitime sertifikater !!!
- Validering av serversertifikat er bare syntaktisk autentisering, ikke semantisk verifisering av netsidens ektehet
- Brukere som ikke kjenner serverens domenenavn, kan på forhånd ikke vite om det er falskt eller ikke

Tillitsforvirring



- Kriminelt-firma er både autentisk og upålitelig/falsk
- For å unngå denne forvirringen må vi være spesifikke om hva vi mener med tillit
- X.509 -sertifikater gir bare garanti om autentisitet, ikke pålitelighet!

PKI sammendrag

- Kryptering med offentlige nøkler trenger en PKI for å være praktisk
- PKI-er er komplekse og dyre i drift
- Internett-PKI er den mest brukte PKI takket være distribusjonen av rotsertifikater med nettlesere
- Sikkerheten til PKI avhenger av integriteten til CA-ene
- PKI-tjenester kalles «tillitstjenester» (Trust Services) i EUs digitale agenda
- PKI og tillitstjenester danner grunnlag for e-ID og e-forvaltning i EU

SLUTT PÅ PRESENTASJONEN