Sluttrapport for UNISA

Norsk Regnesentral
ANVENDT DATAFORSKNING
Norwegian Computing Center/Applied Research and Development

NOTAT/NOTE

OMNI/03/00

Jonn Skretting

Oslo
Desember 2000

UNINETT PCA Certificate

SubjectName:		OU=PCA, O=UNINETT, C=NO
IssuerName:		CN=EuroPKI Root Certification Authority, O=EuroPKI
SerialNumber:		2 (decimal)
Validity -	**NotBefore:**	Wed Jan 5 15:22:13 2000 (000105142213Z)
	NotAfter:	Sat Mar 31 23:59:59 2001 (010331215959Z)
Public Key Fingerprint:		F5BF C381 E8B1 6BA2 6E14 9FEF 8225 EE4F
SubjectKey:		Algorithm RSA (OID 1.2.840.113549.1.1.1), NULL Public modulus (no. of bits = 1024)

Certificate extensions:

Authority Key Identifier:	8CDC 8BB1 A54A 90E7 4E88 7318 3C9D D55E 7EE4 B2CD
Subject Key Identifier:	00BC 9EA2 5382 1CDB 10A7 A8CD 6624 453E 6335 5328
Key Usage:	(CRITICAL) digitalSignature nonRepudiation keyEncipherment dataEncipherment keyCertSign cRLSign
Certificate Policies:	OID 1.3.6.1.4.1.5255.1.1.1 OID 1.3.6.1.5.5.7.2.1OctetString length: 41 octets OctetString: 0 16276874 74703A2F 2F777777 2E657572 |.'http://www.eur| 10 6F706B69 2E6F7267 2F63612F 726F6F74 |opki.org/ca/root| 20 2F637073 2F312E31 2F |/cps/1.1/ |
Subject alternative names:	RFC822: pca@uninett.no
Subject alternative names:	URI: ldap://unisa.nr.no:1522/ou=pca,o=uninett,c=no?cacertificate
Basic Constraints:	allowed to act as a CA !
CRL Distribution Points:	
CRL Distribution Point Names:	URI: http://www.europki.org/ca/root/crl/crl.der
Certificate Fingerprint:	FD:18:DE:F4:5C:FE:E4:C6:67:65:08:1A:57:7D:40:42

Norsk Regnesentral
ANVENDT DATAFORSKNING

NR-notat/NR-Note

Tittel/Title:

Sluttrapport for UNISA

Dato/Date: 15. Desember
År/Year: 2000
Notat nr: OMNI/03/00
Note no:

Forfatter/Author:
Jonn Skretting

Sammendrag/Abstract:
UNISA prosjektet har hatt som målsetning å utvikle en X.509 basert sertifiseringstjeneste for UNINETT. Prosjektet startet høsten 1995 og var et riktig prosjekt til riktig tid. UNISA ble en pionér på sitt område og oppfyller både UNINETTs og NRs ønsker om å kunne evaluere og promotere hensiktsmessig ny teknologi. Vi fikk være med på å generere den første seriøse og offisielle PKI infrastruktur i Norge og var inntil andre kommersielle aktører kom på banen, nesten å regne for et uoffisielt nasjonalt kompetansesenter rundt PKIer og tilhørende bruk av kryptografiske metoder. I løpet av prosjektets fem år har kommersielle aktører fattet stor interesse for sertifiseringstjenester, og vi har fått rammeavtaler for PKIer innenfor forvaltningsnettsamarbeidet.
For UNINETT har det aldri vært noe mål å drive en sertifiseringstjeneste i seg selv. Det er derfor nå naturlig for UNINETT å legge ned sin pilottjeneste og benytte de kommersielle tjenestene som tilbys i markedet også innen UoH sektoren. Det neste naturlige skrittet er så å se på hvordan disse sertifiseringstjenestene kan benyttes for å oppnå de langsiktige målsetningene bak UNISA prosjektet: å kunne tilby integrerte totalløsninger på relevante sikkerhetskritiske områder. På slutten av UNISA prosjektet ble det gjort et forprosjekt for å utrede slike spørsmål. Prosjektet kalles FEIDE (Felles Elektronisk Identitet for UoH sektoren) og egen forprosjektrapport foreligger som oppsummerer dette arbeidet.

Emneord: tiltrodd tredjepart, TTP, offentlig nøkkelsertifikat, offentlig nøkkelinfrastruktur, PKI, sertifiseringshierarki, sertifiseringsautoritet, registreringsautoritet, sikker meldingsutveksling, elektronisk identitet, X.509v3, sertifikatdistribusjon, LDAP, smartkort, Secude

Indexing terms:

Målgruppe/Target group: Personer som har vært involvert i UNISA prosjektet som medarbeidere, brukere eller sponsorer. Personer med generell interesse for sertifiseringsinfrastrukturer.

Prosjektnr/Project no: 28 001

Antall sider/No of pages: 116

Satsningsfelt: Sikkerhet

Research field:

Norsk Regnesentral/Norwegian Computing Center
Gaustadalléen 23, Postboks 114 Blindern, N-0314 Oslo, Norway
Telefon 22 85 25 00, telefax 22 69 76 60

Innhold

Fem år med UNISA

Denne rapporten representerer egentlig sluttrapporten for UNISA prosjektet for de tre siste årene. For årene før det foreligger det en egen sluttrapport [1]. Selv om prosjektet har vært splittet opp i flere påfølgende og administrativt separate kontrakter, så henger det hele sammen faglig og operativt sett som ett stort prosjekt. Utgangen av år 2000 representerer imidlertid slutten på hele prosjektet og det er naturlig ved denne korsvei å se prosjektet under ett; hvor startet vi, hva har skjedd underveis, hvor står vi i dag og hva er hovedkonklusjonen etter prosjektet. Dette dokumentet er på mange måter en ny utgave av [1], men vi vil konsentrere oss mest om overordnede betraktninger og ikke gå så mye i detaljer.

Bakgrunnen for prosjektet kan vi lese om i utredningene for en UNINETT kryptotjeneste våren 1995 [2]:

> I et universitets- og forskningsmiljø er det meste av informasjonen åpen. Det er likevel ikke vanskelig å finne eksempler på informasjon som trenger beskyttelse, men som det er vanskelig å kunne overføre over åpne datanett; eksamensoppgaver og eksamensresultater, utvalgsinnstillinger, administrativ informasjon, personalinformasjon og masse annet. I dag kan ikke slik informasjon overføres, eventuelt skjer dette uten forskriftsmessig beskyttelse. I de fleste tilfeller vil antagelig det viktigste være å sikre mot uautorisert manipulering av informasjon, men ofte er det også ønskelig å forhindre uautorisert innsyn.

Videre sies det at UNINETT på denne bakgrunnen vil etablere en sikkerhetstjeneste med følgende formål:

> Målet med tjenesten er å etablere sikkerhet og sikrede funksjoner for UNINETTs brukere. Det tas sikte på å gi brukerne forholdsvis primitive funksjoner som kan brukes til å sende data via normale UNINETT-tjenester som e-post og FTP, men med en sikkerhet som tilfredsstiller Datatilsynets og Beskyttelsesinstruksens krav til overføringssikkerhet.
>
> Tilsiktet klassifikasjon av data er Begrenset/Fortrolig; det forventes ikke at disse verktøyene er gode nok for Strengt Fortrolig eller Hemmelig. Det fokuseres på de rent overføringsmessige aspekter, journalføring eller tilsvarende er ikke inkludert.

For å realisere tjenesten ble det så besluttet å etablere en sertifiseringstjeneste kalt UNISA (*UNInett SertifiseringsAutoritet*) med sertifiseringshierarkier og en offentlig nøkkel infrastruktur (PKI, *Public Key Infrastructure*) og med tilhørende brukerprogramvare for signering og kryptering av meldinger. UNINETT ble på denne tiden også med i EU prosjektet ICE-TEL som hadde som formål å etablere en slik felles infrastruktur for hele Europa og med en felles europeisk rot. UNISA arbeidet kan på mange måter sees på som UNINETTs bidrag til ICE-TEL.

Konkret innhold i prosjektet

NR inngikk en kontrakt med UNINETT om utvikling og drift av UNISA tjenesten høsten 1995. De første årene ble det naturlig nok mye utvikling, for slike infrastrukturer eksisterte ikke fra før av (i Norge i hvertfall) og UNISA ble en pionér på sitt område. Den første PKIen som ble generert var koplet opp mot IPRA (*Internet Policy Registration Authority*) og var en X.509v1 infrastruktur[1]. Den ble seinere erstattet av en X.509v3 infrastruktur med ICE-TEL rot administrert av UNI-C i Danmark[2]. UNINETT var på mange måter den fremste partneren i ICE-TEL når det gjaldt å få etablert en infrastruktur, og flere andre partnere bygget videre på våre erfaringer. Ved slutten av ICE-CAR prosjektet (etterfølgeren til ICE-TEL) så administreres roten nå av Politecnico di Torino. PKIen har fått navnet EuroPKI (`http://www.europki.org/`) og rot-sertifikatet er generert med en levetid på ti år; det utløper 31. Desember 2010.

Stikkord for arbeid i UNISA har vært:

- teknologiintegrasjon og -evaluering (bl.a. tilpasning av sluttbrukerverktøy som e-post lesere og nettlesere på ulike plattformer og evaluering av nye versjoner av SECUDE),
- uttesting av tekniske løsninger,
- utarbeidelse av sikkerhetsinstrukser i form av policier og såkalte *practice statements* (vår aller første policy ble gitt ut som RFC 1875 [15]),
- markedsføring av tjenesten mot potensielle brukere,
- brukerstøtte og rådgivningstjenester (inkludert web-sider),
- operativ drift (utstedelse og distribusjon av sertifikater og tilbakekallingslister),
- prosjektsamarbeidet i ICE (forkortelse for ICE-TEL *og* ICE-CAR) som sådan.

Vi har ved utgangen av år 2000 en tjeneste som har kjørt nokså uendret i i hvertfall et år. Overordnet struktur på denne tjenesten er beskrevet i avsnittet *"Overordnet beskrivelse av UNISA"* og mer detaljert driftsdokumentasjon finnes i Vedlegg F. Policy og practice statements finnes i Vedlegg B og Vedlegg C, henholdsvis.

NR har drevet de fleste sertifiseringsautoritetene (CAene) på vegne av høgskoler og universiteter, men et par institusjoner (Høgskolen i Agder og Høgskolen i Østfold) var i perioder så interesserte i UNISA at de valgte å drive CAen sin selv. Høgskolen i Østfold var vel den mest aktive av dem og markedsførte UNISA i en egen artikkel i HØit [18].

Når det gjelder prosjektsamarbeidet i ICE så fikk vi gjennom dette først og fremst deltakelse i et faglig forum som arbeidet akkurat med det samme som vi var interessert i selv, og vi fikk tilgang til nødvendig programvare for faktisk å kunne realisere den tjenesten vi så for oss. Ettersom verktøyutviklerne var med i ICE, så hadde vi også rikelig med muligheter for å kunne påvirke utviklingen av verktøyene i den retningen vi ønsket oss og få implementert den funksjonaliteten som til enhver tid var viktigst for oss. Hovedverktøyet vårt har vært SECUDE som opprinnelig kom fra GMD, men som nå er fullt kommersialisert og eies og videreutvikles av firmaet SECUDE som ble stiftet i prosjektperioden.

1) Det første UNINETT PCA sertifikatet fra IPRA var datert 21.09.1995. Det hadde en gyldighetesperiode på ett år og ble fornyet én gang.
2) UNINETT PCA var sertifisert av denne roten i tidsrommet fra 14.11.1997 til 31.12.1999.

Markedsføring av tjenesten har skjedd ved presentasjoner og stander (med for anledningen spesiallagede plakater) på UNINETT konferanser i 1995 [4], 1996 og 1997 [5] og NORDUnet konferansen i 1998 [6]. Der har vært artikler og meldinger i UNINytt ([16], [17]) og direkte henvendelser til driftspersonale ved universiteter og høgskoler og andre relevante institusjoner som blant annet Statens Lånekasse for utdanning. Aktuelle markedsføringsreferanser i form av foredrag er [7] [8] [9] [10] [11] [12] [13]. Vi hadde besøk av, og kjørte demo av tjenesten for, en journalist fra *Computerworld* som så skrev en i øynefallende artikkel med bilde av prosjektdeltakerne i neste nummer av bladet [19]. Der var også en populærvitenskapelig artikkel om *Sikker utveksling av e-post* i Teknisk Ukeblad [14], og *News* har vært benyttet. UNISAs nettsted har i prosjektperioden vært: `hhtp://www.uninett.no/pca/` og web-sidene har vært en hovedkanal utad. Innfallsportene `www.uninett.no/pca/index.html` og `www.uninett.no/pca/index-e.html` ble fra 01.01.99 til 13.12.99 aksessert tilsammen 4.316 ganger. Fratrukket er da aksessene fra maskinen som ble benyttet som hovedprosjektmaskin på NR.

Som et ledd i å få promotert UNISA tjenesten har vi i tillegg tilbudt en PGP (*Pretty Good Privacy*) tjeneste. UNISA genererte en nøkkel i UNINETTs navn og tilbød å signere nøklene til utvalgte personer som så måtte forplikte seg (jmf. Vedlegg D) til å følge til UNINETTs policy (jmf. Vedlegg E) med hensyn på videre signering og beskyttelse av sin egen private nøkkel. Policy ble offentliggjort, og personer med nøkler signert direkte av UNINETT ble oppgitt, på web-sidene til UNISA. Tjenesten ble tilbudt ettersom vi visste at det fantes en del PGP brukere. Fordelen med PGP er at det er lett å komme i gang. Vi håpet at vi etterhvert kunne få PGP brukerne til også å benytte X.509 tjenesten og ellers øke bevisstheten til sikker meldingsutveksling blant brukerne generelt. Men PGP tjenesten ble aldri noen stor suksess og den ble nedlagt pr. 1. Januar 2000.

Vi har også tilbudt en demotjeneste for utstedelse av sertifikater. Denne har vært tilgjengelig fra web og kommer opprinnelig fra SECUDE/GMD. Vi gjorde våre tilpasninger til norske forhold. Der kunne alle sende en sertifiseringsforespørsel som så ble håndtert automatisk uten noen form for identitetssjekk. Formålet med demotjenesten skulle være å gi brukerne et tilbud om testsertifikater som de kunne eksperimentere med før de eventuelt bestemte seg for å få et ordentlig sertifikat.

Problemet i UNISA har vært å få brukere av tjenesten og pilotprosjekter. I mangel av piloter så har vi strengt tatt ikke høstet andre erfaringer rundt praktisk bruk av infrastrukturen enn de vi har gjort selv. Det er beklagelig at det ikke er blitt gjennomført større piloter slik at vi kunne forsket litt på hvordan teknologien fungerer i praksis. Hvor er det så at skoen trykker? Vi har noen teorier og har gjort noen erfaringer:

- brukerne ser ikke noe prekært behov, og potensielle brukere har lite tid til ikke-prekære gjøremål,
- det holder ikke med enkeltbrukere, vi må ha flere brukere som har behov for å kommunisere sammen om noe,
- problemer med meldingsformater,
- PKIens rotsertifikat leveres ikke som en del av Netscape og Microsoft Internet Explorer.

Å sertifisere enkeltbrukere har lite for seg om ikke de brukerne som de har behov for å kommunisere med også blir sertifisert. Derfor prøvde vi å identifisere en gruppe av brukere som kunne ha interesse av å sikre kommunikasjonen seg i mellom. Tjenesten ble forsøkt solgt til Samordnet Opptak som helt klart hadde et behov og som også var interessert. Til

tross for at vi hadde flere møter med dem så ble det aldri til at de tok UNISA tjenesten i bruk. De var angivelig veldig presset på tid og det vil klart kunne være ulike vurderinger knyttet til hvor prekært det var/er for dem å få bedret sikkerheten.

Jo flere sertifikater som er utstedt, desto større blir anvendeligheten og dermed verdien av, hvert enkelt sertifikat. Et poeng som kanskje symboliserer at det ikke er sikkerhet i seg selv som er uinteressant for brukerne, men mer det at "det er ikke noe poeng i å være den eneste som er sertifisert", er det faktum at de sertifikatene vi har utstedt til personer utenfor UNINETT og NR i all hovedsak har vært web-server sertifikater. Det kommer trolig av det faktum at for å oppnå sikker kommunikasjon på SSL nivå med en web-server så holder det at serveren er sertifisert, klienten kreves det ikke noe av annet enn at den kan snakke SSL. Dermed får ett sertifikat alene ganske stor betydning.

Når det gjelder meldingsformater så finnes det egentlig ikke noe felles som kan benyttes overalt. PEM (*Privacy Enhanced Mail*) ble benyttet de første årene av prosjektet og ble av ICE-TEL partnere integrert i Eudora og Outlook. UNINETT laget integrasjon for Exmh. Nå må PEM kunne sies å være død, og det er egentlig S/MIME som benyttes. Problemet her har vært at det kun har vært tilgjengelig på Microsoft platform og ikke på Unix bortsett fra gjennom Netscape. Nå er det kommet en S/MIME utility for OpenSSL som forhåpentligvis kan være til hjelp. SECUDE tilbyr PKCS#7 kommandoer, og resten må man gjøre selv.

Når det gjelder at rotsertifikatet ikke har vært tilgjengelig som standard vedlegg i kjente nettlesere, så er dette noe ICE har prøvd å gjøre noe med. Men svaret har enten vært at dette med å inkludere CAer i nettleserne var noe leverandørene skulle slutte med for de ønsket ikke å ta slike sikkerhetsavgjørelser på vegne av sine brukere, til en angivelse av en pris som lå skyhøyt over hva ICE ville kunne ha muligheten for å tilby. Fordelen med et rotsertifikatet inkludert i nettleseren er at brukerne ikke selv må oppsøke UNISAs nettsted for å laste det ned eller evt. ikke må gå gjennom en godkjennelsesprosess hver gang et sertifikat utstedt av UNISA mottas fra en server. Nedlasting er i seg selv en liten operasjon[3] og kan karakteriseres som en bagatell, men det tillegges likevel stor betydning. Både representanter for UiB og UiO vurderte f.eks. på denne bakgrunnen UNISA som en lite attraktiv tjeneste, og UiO valgte *Thawte*.

Hvor står vi?

UNINETT har besluttet å avslutte UNISA pilottjenesten fra og med 01.01.2001 samtidig med at ICE-CAR prosjektet avsluttes. Dette skyldes da at sertifiseringstjenesten som sådan er lite etterspurt blant UNINETTs medlemsorganisasjoner og dessuten at tilsvarende tjenester nå kan kjøpes kommersielt i markedet om nødvendig.

Når det gjelder kommersielle aktører så finnes det flere, og noen vil utvilsomt ønske å posisjonere seg i UoH markedet. Det vil de kunne få til ved f.eks. å inngå et samarbeid med Studentsamskipnaden i Oslo (SiO) som har gående et pilotprosjekt med studiekort med smartkortfunksjonalitet. SiO ser et stort potensiale i smartkortteknologien og ønsker

3) Brukeren trykker på en knapp for nedlasting og går så gjennom en dialog hvor sertifikatet tilslutt aksepteres. Dette er den samme dialogen som brukeren vil måtte gå gjennom dersom han får presentert et sertifikat hvor nettleseren ikke gjenkjenner den CAen som har signert sertifikatet.

å utvide dagens funksjonalitet i kortene. Et høyst aktuelt tillegg er elektronisk ID. Slik elektronisk ID kan da f.eks. benyttes til meldingssikkerhet.

Det skal så sies at det aldri har vært noe mål for UNINETT å drive en operativ sertifiseringstjeneste i seg selv. Det er naturlig å terminere tjenesten ettersom den nesten ikke har brukere, men samtidig kunne det også vært aktuelt å sette den ut om den hadde hatt veldig mange brukere, for det å betjene en slik struktur utover som en pilottjeneste krever mye administrasjon og ressurser om det skal skje i henhold til en akseptabel policy. Det viktigste er at tjenestene er tilgjengelige i markedet (og gjerne norske tjenester) for UNINETTs brukere.

ICE prøvde å etablere en felles europeisk PKI ved nærmest å bygge PKIen ovenfra og ned, og det gjenstår å se om EuroPKI faktisk vil lykkes. Et alternativt, og kanskje mer realistisk scenario, er selvstendige PKIer som etableres relativt lokalt og som så sammenkoples med andre PKIer ved en form for kryss-sertifisering når åpenbare økonomiske gevinster finnes som beveggrunn i hvert enkelt tilfelle.

Om vi ser tilbake på formålet med UNINETTs sikkerhetstjeneste som referert innledningsvis, så stod det der: *"Det tas sikte på å gi brukerne forholdsvis primitive funksjoner som kan brukes til å sende data via normale UNINETT-tjenester som e-post og FTP, …"*. Denne basistjenesten har vist seg ikke å være tilstrekkelig attraktiv isolert sett. Men selve bakgrunnen for prosjektet (som også angitt innledningsvis) er der fremdeles likefullt. Og UNINETT vil fortsatt ha et perspektiv på anvendelsesområder for elektronisk identitet og meldingssikkerhet for å tilfredsstille de behovene som angis i bakgrunnen for prosjektet. Men det vil finne en annen form enn å være en ren sertifiseringstjeneste. I 1995 var det naturlige skrittet å tilby en sertifiseringstjeneste. I dag er det naturlige neste skrittet å forsøke å tilby mer integrerte totalløsninger for de anvendelsesområdene som skisseres. Dette har da også UNINETT allerede begynt å se på gjennom sitt FEIDE initiativ. Rapporten fra FEIDE forprosjektet [3] detaljerer visjonene selv i innledningen. Nå vil nok også FEIDE prosjektet kunne ha utfordringer når det gjelder å finne passende piloter, men la oss håpe at de vil lykkes med det.

Hovedkonklusjon

UNISA var et riktig prosjekt til riktig tid. Det oppfyller både UNINETTs og NRs ønsker om å kunne evaluere og promotere hensiktsmessig ny teknologi. Vi har fått være med på å generere den første seriøse og offisielle PKI infrastruktur i Norge og var inntil andre kommersielle aktører kom på banen, nesten å regne for et uoffisielt nasjonalt kompetansesenter rundt PKIer og tilhørende bruk av kryptografiske metoder. (Web-sidene til prosjektet har f.eks. vært å se på NRK Kveldsnytt i forbindelse med et innslag om datamaskiner og sikkerhet.) Prosjektdeltakere har siden kunnet benytte sine erfaringer fra UNISA i forbindelse med arbeidet både i Rådet for IT-sikkerhet og rundt arbeidet med evaluering av samarbeidsavtalene i Forvaltningsnettsamarbeidet, såvel som i en rekke andre sammenhenger som det ville føre for langt å komme inn på her. Selv om vi gjerne skulle hatt noen piloter, så vil vi likevel ubetinget karakterisere UNISA som et vellykket prosjekt.

Overordnet beskrivelse av UNISA

Det sertifiseringshierarkiet som eksisterer på slutten av prosjektet, presenteres nedenfor. Deretter sier vi noe om det driftsmiljøet som har vært nødvendig for å kunne håndtere sertifiseringshierarkiet. Til slutt sier vi kort noe om de viktigste operasjonene sett fra en brukers synsvinkel: nøkkelgenerering, sertifisering, signering og kryptering og litt mer om sikker meldingsutveksling i e-post og web sammenheng.

Dette avsnittet er en oppdatering av tilsvarende informasjon i [1].

Sertifiseringshierarki

UNISA er basert på bruk av offentlige og private nøkler, såkalt asymmetrisk nøkkelteknologi, som muliggjør bruk av digitale signaturer og kryptering. Teknologien forutsetter imidlertid at en oppgitt kopling mellom en innehaver og vedkommendes offentlige nøkkel alltid er korrekt og uforfalsket. Denne koplingen stadfestes i UNISA ved hjelp av tiltrodde tredjeparter (TTPer), eller sertifiseringsautoriteter (CAer), som utsteder og signerer offentlig nøkkelsertifikater for brukere av tjenesten. Sertifikatene benyttes så for kryptering av meldinger og verifikasjon av digitale signaturer.

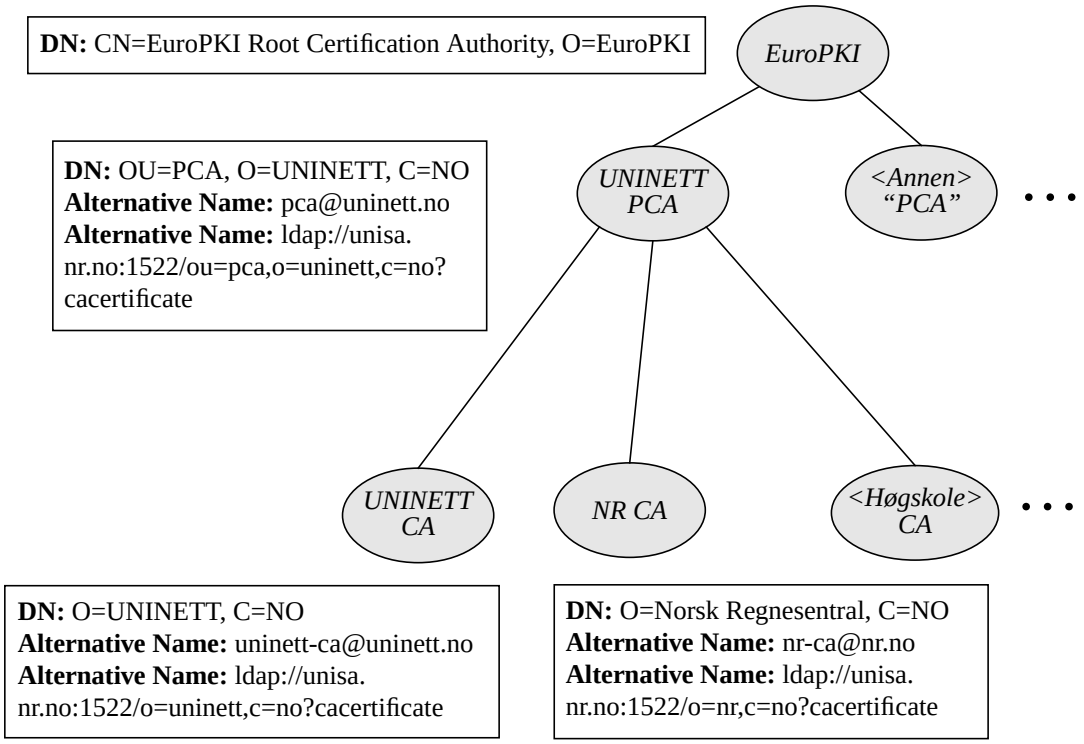
En sertifiseringsinfrastruktur består av en eller flere sertifiseringsautoriteter der hver enkelt CA opptrer som en tiltrodd tredjepart som vi kan velge å stole på. En sertifiseringsautoritet utsteder sertifikater, hvor sertifikatene er et identitetsbevis i elektronisk form. En sertifiseringsautoritet kan utstede et sertifikat for en annen CA, og dermed være tiltrodd tredjepart for denne CAen, og slik kan vi skape en infrastruktur av tillit som kan verifiseres. Et sertifikat må ha et standard format som alle kan gjenkjenne for at det skal kunne brukes i ønskede sammenhenger. Et slikt akseptert format for sertifikater er X.509 standarden (siste og mest brukte versjon er versjon 3[1]) som sier noe om hva sertifikatet skal inneholde. X509v3 er nærmere beskrevet i Vedlegg A.

For at vi skal kunne avgjøre hvorvidt vi ønsker å stole på en CA eller ikke, så må CAen si noe om nivået på den sikkerheten som ligger i identitetsbeviset el. sertifikatet. En CA har derfor muligheten til å utgi en policy, også kalt sikkerhetsinstruks, som sier hvilke organisatoriske og tekniske krav som CAen følger. Enhver CA må i prinsippet følge den policien som den overordnede CAen har utstedt og har selv kun anledning til å forsterke denne. I ICE-CAR/EuroPKI sammenheng endte vi opp med en modell der kun roten utsteder en policy. Underordnede CAer utsteder så *practice statements* (en beskrivelse av de konkrete tiltakene som iverksettes for å implementere policien i praksis). Slike practice statements

1) Versjon 3 åpner for et større anvendelsesområde og bedre brukervennlighet enn det som var mulig med versjon 1.

må godkjennes av den CAen som har utstedt policien. Begrunnelsen for å ha kun én policy for hele PKIen er å slippe å måtte verifisere at en policy er like sterk som en annen. Dette er en tung operasjon, det er lettere å verifisere practice statements. EuroPKI sin policy er gjengitt i Vedlegg B og UNINETT sine practice statements for CAen i EuroPKI (kalt UNINETT PCA[2]) er gjengitt i Vedlegg C.

ICE/EuroPKI sertifiseringsinfrastrukturen som UNINETT PCA har vært en del av er illustrert i Figur 1.



Figur 1: CA hierarkiet i ICE-TEL og UNISA

Ved slutten av prosjektet har UNINETT PCA sertifisert CAer for UNINETT, NR, Universitetet i Tromsø, Universitetet i Bergen og Høyskolen i Oslo. I tidligere versjoner av PKIen var også andre institusjoner med, deriblant Høyskolen i Agder og Høyskolen i Østfold som drev sine egne CAer (alle andre CAer har vært drevet av NR i prosjektperioden). Tilsammen ved slutten av prosjektet er det under UNINETT PCA utstedt 11 SECUDE sertifikater, 7 web klient sertifikater og 5 web server sertifikater[3].

For demonstrasjonstjenesten *TrustFactory* er det utstedt tilsammen 84 sertifikater; 60 webserversertifikater og 24 klientsertifikater. Disse er imidlertid utstedt til brukere over hele verden og ikke bare i Norge, *TrustFactory* tjenesten er tilgjengelig for alle som finner fram til den.

2) Navnet burde vært UNINETT CA eller noe lignende. PCA (Policy Certification Authority) er noe som henger igjen fra et tidligere stadium da CAen utstedte sin egen policy.

3) I tidligere versjoner av PKIen der det ble gjort massesertifisering, så var antallet utstedte sertifikater betydelig høyere (men den totale bruken av sertifikatene var ikke nødvendigvis større).

Registreringsautoriteter

Det er essensielt at koplingen mellom brukeres identitet og offentlige nøkler, slik det fremgår i sertifikatet, er tilstrekkelig verifisert. Brukere må derfor presentere godkjent legitimasjon og bekrefte sin offentlige nøkkel ved utstedelse av ethvert sertifikat. CAer vil imidlertid kunne være plassert langt fra brukerne, og vi har derfor innført rollen registreringsautoritet (RA). En RA er en betrodd representant fra en organisasjon som er autorisert av CAen til å foreta identitetskontroll på vegne av CAen. En bruker vil i dette tilfelle kunne legitimere seg og bekrefte sin offentlige nøkkel lokalt hos RAen, og RAen vil i sin tur signere informasjon om identitet og nøkkelbekreftelse og sende dette til CAen. Brukeren vil i tillegg sende en normal sertifikatforespørsel til CAen.

Driftsmiljø

I dette avsnittet sier vi litt om det driftsmiljøet som er blitt brukt for å håndtere sertifiseringstjenesten.

Programvare

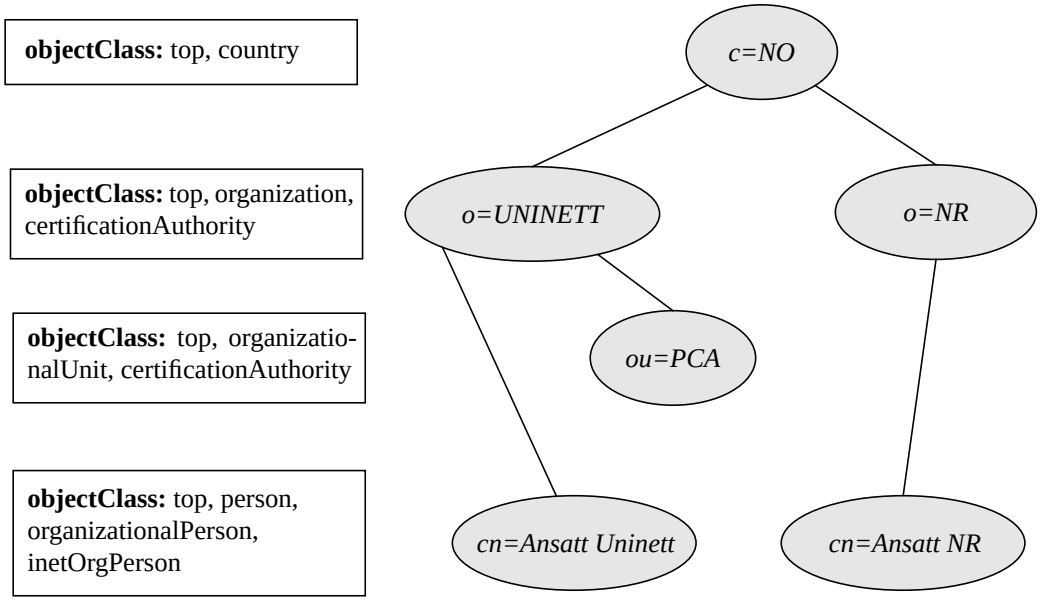
For utstedelse og administrasjon av sertifikater, samt opprettelse av CAer, har programvaren SECUDE fra SECUDE/GMD vært benyttet. Programvaren kan også benyttes av sluttbrukere for administrasjon av sertifikater, og som basis for sikker e-post. Vi har benyttet SECUDE versjon 5.2b. Siste versjon av SECUDE er i skrivende stund 5.4.17. Vi har imidlertid ikke hatt behov for noen nyere versjon enn 5.2b og har benyttet denne. SECUDE er tilgjengelig på alle aktuelle plattformer.

Siden UNISA bygger på standarden X.509v3 for sertifikatformat, så kan en bruker i prinsippet velge å benytte hvilken som helst programvare som støtter X.509v3. Interoperabilitetstester er blitt utført, og det skal være rimelig greit for brukere med forskjellig programvare å kunne kommunisere med hverandre. UNISA legger ingen begrensninger på brukeren når det gjelder hvilken programvare som skal anvendes. Dette innebærer spesielt at en bruker kan velge å bruke krypto-programvare innebygget i de applikasjonene som tilbyr dette.

Informasjon om tilgjengelig krypto-programvare har vært tilgjengelig på UNISAs websider. Også såkalte plug-in moduler har vært tilgjengelige for e-post applikasjoner som Eudora, Exchange, Z-mail og Exmh. Disse modulene ga støtte for PEM, men i og med at PEM er mer eller mindre ute, så har de ikke lengre like stor verdi.

LDAP[4] serveren *Enterprise Directory Server* (EDS) fra Messaging Direct (Isode Ltd) har vært benyttet for å distribuere sertifikater og tilbakekallingslister. URLen har vært `ldap://unisa.nr.no:1522/c=no`. DITet (*Directory Information Tree*) som har vært benyttet er illustrert i Figur 2. CA sertifikater har vært lagt i attributtet `cACertificate` og tilbakekallingslister i `certificateRevocationList`. Brukernes sertifikater til benyttelse for S/MIME meldinger har vært lagret i `userSMIMECertificate` og andre sertifikater i `userCertificate`.

4) Lightweight Directory Access Protocol, RFC2251



Figur 2: Prinsipper for UNISA LDAP DIT

Web server sertifikater har ikke vært distribuert med LDAP i UNISA. Policien som sier at absolutt alle sertifikater må distribueres med mindre at eieren eksplisitt motsetter seg det, ble først vedtatt helt på slutten av ICE-CAR. Motivasjonen bak ikke å distribuere web-server sertifikater vhja. LDAP har vært at sertifikatene benyttes i SSL sammenheng og SSL protokollen spesifiserer at en web-server alltid skal svare med sitt sertifikat når den kontaktes av en klient. Derfor skulle det ikke være nødvendig for noen å hente et web-server sertifikat fra en egen distribusjonstjeneste. Når det så allikevel er tatt inn i policien, så er det av generelle hensyn og for å gjøre det mulig for de som vil se på sertifikatet å hente dette fra det samme stedet hvor alle andre sertifikater finnes.

Dokumentasjon

Brukere av UNISA har som nevnt behov for å vite sikkerhetsnivået på tjenesten for å kunne avgjøre hvilken tillit de skal ha til den. Det er derfor nødvendig å få beskrevet alle rutiner og det tilsier at alle organisatoriske og tekniske forhold ved UNISA må være dokumentert. Med unntak av driftsrutinene for CAene så har dokumentasjonen i sin helhet vært tilgjengelig på UNISAs web sider.

Dokumentasjonen av organisatorisk art omhandler:

- policyer
- practice statements
- instrukser
- retningslinjer
- avtaler og
- driftsdokumentasjon.

NR har aktivt bidratt til EuroPKI policien (som er sterkt influert av et utkast til policy forfattet av NR), samt skrevet tre practice statements som omhandler utstedelse av vanlige

sertifikater og sertifikater for web klienter og web servere. Dokumentasjonen henvender seg både til sluttbrukere og systemadministratorer, samt CAer og RAer.

Den tekniske dokumentasjonen tar i første rekke for seg

- installasjon
- konfigurasjon og
- bruk

av SECUDE programvaren, X.509v3 sertifikater som sådan, Exmh og Z-Mail klienter, Eudora integrasjonsmoduler og LDAP tjenesten. Visse aspekter av SSL er omhandlet, og dokumentasjonen dekker også en låsesmedtjeneste. Sistnevnte har vært et tilbud til brukere for å sikre at de skal kunne nå sine private nøkler dersom de eventuelt skulle glemme sin passordfrase. UNISA sin web dokumentasjon omfatter også et:

- innføringskurs i kryptografi og nøkkeladministrasjon.

Generell dokumentasjon av SECUDE programvaren har vært tilgjengelig på web fra SECUDE.

Smartkort

Dersom en CAs private nøkler skulle bli kompromittert, så ville sikkerheten i store deler av tjenesten være truet. Som en tilleggssikkerhet, og for uttestingsformål, har UNINETT PCAens private nøkler vært lagret på TCOS smartkort i ICE-CAR prosjektperioden. Gjennom bruk av SECUDE har imidlertid også sluttbrukere hatt muligheter for å benytte TCOS smartkort om de så ønsket.

Sertifikatdistribusjon og tilbakekallingslister

UNINETT fikk konsesjon fra Datatilsynet til å opprette et sertifikatregister for UNISA. Sertifikatregisteret har vært offentlig tilgjengelig og fire tjenester for søk etter sertifikater er blitt brukt brukt i prosjektperioden; én basert på X.500, én på LDAP, én på e-post og én på bruk av web. Men alle har ikke vært tilgjengelige samtidig. X.500 ble bare benyttet helt innledningsvis før vi gikk over til å lagre sertifikater kun på filer og filkataloger. Først det siste året har vi lagret sertifikatene i en LDAP server. Da LDAP serveren ble introdusert, så ble den e-post baserte søketjenesten lagt ned. Og siden LDAP katalogen ble opprettet så har web-distribusjonen benyttet CGI Perl scripts med PerlDAP for å hente fram sertifikater og tilbakekallingslister.

Maskiner

Alle CAer har ligget på en dedikert og ikke nettilkoplet maskin innelåst på eget rom med adgang kun for nøkkelpersonell. En SPARC maskin med SunOS 4.1.3 har vært benyttet. Den har hatt tilkoplet en SNI smartkortleser for å kunne aksessere TCOS smartkort. Da PGP tjenesten var operativ så ble også denne administrert fra en dedikert PC innelåst på samme rom som CA maskinen.

Bruksaspekter

Der er fire overordnede operasjoner som er relevante for alle brukere; nøkkelgenerering, sertifisering, signering og kryptering. Dessuten må brukerne overholde de sikkerhetsinstruksene som gjelder for sertifiseringstjenesten hva angår beskyttelse av private nøkler.

Nøkkelgenerering

UNISA anbefaler brukerne å generere nøkler selv. Dermed kan brukeren føle seg helt sikker på at ingen andre har kopi av den private nøkkelen, så lenge han/hun stoler på nøkkelgenereringsprogrammet. Nøkkelgenerering og sertifisering skal utføres kun én gang av brukeren; i det brukeren ønsker å ta tjenesten i bruk. Automatisk nøkkelgenerering tilbys av de fleste kryptoprogrammer, og den private nøkkelen vil bli lagret kryptert på brukerens lokale disk eller smartkort. Brukeren trenger aldri å se den private nøkkelen, men vil måtte angi et passord for å aksessere den. Vanligvis angis passordet i den applikasjonen hvor brukeren ønsker å benytte nøkkelen.

UNISA tilbyr et web grensesnitt for automatisk nøkkelgenerering for web klientapplikasjoner. I tillegg genererer integrasjonsmoduler for e-post programmer automatisk nøkler gjennom enkle menyvalg.

UNISA vil også kunne tilby generering av nøkler for en gruppe av brukere om en organisasjon ønsker dette. Organisasjonen vil da være ansvarlig for å distribuere nøklene som er generert på en sikker måte til de riktige personene, og installere nøklene slik at de kan bli tatt i bruk.

Sertifisering

Etter at brukeren har generert nøkler, vil han/hun måtte skaffe seg en sikker kopling mellom identitet og nøkkel gjennom et sertifikat. Et sertifikat er et elektronisk identitetsbevis som beviser hvem nøkkelen tilhører. Sertifikatet utstedes av UNISA tjenesten ved brukerens sertifiseringsautoritet etter at brukeren har vist fram legitimasjon som f. eks. førerkort eller bankkort. Brukeren må derfor oppsøke organisasjonens kontaktperson (registreringsautoritet) og bevise sin identitet og sitt eierskap til nøkkelen. Brukeren sender selv en forespørsel om utstedelse av sertifikat til sertifiseringsautoriteten.

UNISAs web grensesnitt for nøkkelgenerering støtter det å sende en forespørsel om sertifikat til sertifiseringsautoriteten. Det samme gjør plug-in moduler for e-post programmer. Etter at identitet er fremvist, vil brukeren få tilsendt sertifikatet og kan installere dette gjennom kommandoer i den applikasjonen brukeren benytter.

UNISA legger ingen formelle begrensninger i hva brukeren kan anvende sertifikatet sitt til. Det er opp til de som anerkjenner sertifikatet å bestemme hva brukeren kan oppnå ved å fremvise den elektroniske legitimasjonen.

Signering

Ved signering anvender brukeren sin private nøkkel for å beskytte informasjonen som skal utveksles. Informasjonen beskyttes mot at noen skal kunne endre innholdet underveis uten at mottakeren har mulighet for å oppdage det. Om informasjonen er signert og noen endrer innholdet underveis, så vil en mottaker kunne oppdage dette når meldingen mottas.

I tillegg legges typisk andre data ved meldingen, bl.a. sertifikatet, som godtgjør at det er den angitte avsenderen som har sendt informasjonen. Mottakeren vil dermed få autentisert identiteten til avsenderen på en sikker måte.

Kryptering

Ved å kryptere vil avsenderen sørge for at kun mottakeren kan lese innholdet av informasjonen som utveksles. Avsenderen bruker informasjon i mottakerens sertifikat for å kryptere og det er kun mottakeren som kan dekryptere. Avsender må skaffe seg mottakerens sertifikat på forhånd, f.eks. gjennom mekanismer for sertifikatdistribusjon som tilbys av UNISA.

Som oftest vil en kryptert melding også være signert.

Sikkerhetsinstrukser

Følgende krav er satt til sluttbrukere av sikkerhetstjenesten:

- Passord til en brukers PSE (Personal Security Environment) skal bestå av flere ord (passord*frase*). Det forventes at "skikk og bruk" følges ved valg av passord (kombinasjon små/store bokstaver osv.)
- Nøkkelen eller passordet skal under ingen omstendigheter lagres i klartekst noe sted.
- Ved mistanke om at passord og/eller den private nøkkelen er kompromittert skal det øyeblikkelig meldes fra om dette til lokal sertifiseringsautoritet og til de som en vet stoler på sertifikatet (de som en sender signerte/krypterte meldinger til).
- For å kunne ha noen egentlig formening om hvilken sikkerhet som er knyttet til bruken av et sertifikat, så må alle brukerne kjenne til gjeldende policy og tilhørende practice statements.

For systemadministratorer har følgende tillegg også vært eksplisitt poengtert:

- For RAer gjelder samme instruks som for sluttbrukere pluss at RAene må sette seg grundig inn i gjeldende practice statements.
- CAer må sette seg grundig inn i gjeldende policy og alle gjeldende CPSer.

Meldingsformidling

Vi har antatt at et flertall av brukerne vil bruke sertifikatene sine for å oppnå sikker e-post og sikker web kommunikasjon. Det er likevel ikke noe problem å benytte signering og kryptering til å beskytte informasjon som sendes f. eks. via *ftp*. Generelt sett kan brukerne anvende nøkler og sertifikater til å beskytte informasjon, hva som så gjøres med den beskyttede informasjonen er opptil brukerne selv å bestemme.

All utveksling av meldinger tilsier at brukerne må utveksle legitimasjon for å kunne kommunisere sikkert med hverandre. Dette gjøres ved at brukerne får tilgang til hverandres sertifikater. Siden sertifikater er offentlig informasjon så kan de distribueres gjennom alle de elektroniske kanalene som er tilgjengelige. Den hittil vanligste og mest komfortable måten å utveksle sertifikater på er at en avsender sender en e-post melding i klartekst til en mottaker og legger ved sertifikatet i meldingen. Mottakeren kan da lagre sertifikatet

lokalt for seinere bruk. Andre metoder er automatisk oppslag av programvaren som brukes, for å hente ned sertifikatet fra en bestemt plass. Fra UNISAs web sider har det vært mulig å finne sertifikater ved å oppgi en brukers e-post adresse. Ved å "klikke" på brukerens e-post adresse, vil sertifikatet automatisk lastes ned. Det har også vært mulig å finne sertifikatene basert på deler av e-post adressen og utifra organisasjonstilhørighet. Jmf. ellers det tidligere avsnittet *"Sertifikatdistribusjon og tilbakekallingslister"*.

Sikker e-post

Signerte og krypterte meldinger må inneholde en del nødvendig ekstra-informasjon for at meldingen skal kunne bli forstått og verifisert hos mottakeren. UNISA tilbyr støtte for to forskjellige meldingsformater, som begge kan brukes både for signering og kryptering. Disse formatene er PEM, som står for Privacy Enhanced Mail, og S/MIME som står for Secure Multipurpose Internet Mail Extensions.

PEM beskriver et rent tekstformat hvor meldingen kun kan inneholde tekst, og hvor sertifikat og algoritmeidentifikatorer legges ved meldingen innenfor en bestemt tekstbasert syntaks. Mottakeren må forstå PEM syntaksen for å kunne verifisere en signert melding, men vil likevel kunne se hele meldingen som bare tekst hvis han/hun ikke skjønner PEM. S/MIME beskriver et meldingsformat som også støtter ikke-tekstbasert innhold og vedlegg som er ikke-tekstbaserte, og hvor sikkerheten legges ved meldingen i bestemte blokker som angir algoritmer og sertifikat. Mottakeren vil måtte forstå S/MIME syntaksen for å kunne verifisere signaturen i en melding, men vil som for PEM kunne se selve meldingen hvis han/hun ikke har et e-post program som kan tolke S/MIME. Hvis innholdet av en melding er kryptert og de vedlagte sikkerhetsdatene ikke kan tolkes, så vil vi verken i PEM eller S/MIME få fram forståelig informasjon. Begge meldingsformatene er standardiserte og det finnes applikasjoner og plug-ins for applikasjoner, som støtter formatene.

Brukeren anbefales å ha kryptoprogramvare integrert i e-post verktøyet for enkelt å kunne signere og kryptere e-post meldinger. Noen e-post applikasjoner har innebygget støtte for kryptering og signering, og støtter dermed minst ett av meldingsformatene nevnt ovenfor.

Sikker web kommunikasjon

Ved å utstede et sertifikat og installere det i en web server vil vi få det vi kaller en *sikker* web server. En sikker web server vil alltid autentisere seg overfor klienter som kontakter den. Klientene kan så velge om de vil akseptere serverens identitetsbevis eller ikke. Dersom sertifikatet aksepteres og det regnes for å være et solid identitetsbevis, så kan klienter være tryggere på at den serveren de har kontakt med virkelig er den serveren de tror, og så i tur evt. ha større tiltro til det de henter ned fra serveren, som f.eks. programvare.

En sikker server tilbyr kryptering av kommunikasjonen mellom klient og server ved å bruke protokollen SSL (Secure Socket Layer) uten at klienten trenger å ha et sertifikat. Dette vil hindre uautorisert innsyn i de dataene som sendes mellom klient og server, noe som er viktig hvis det f.eks. sendes passord eller brukernavn over kommunikasjonslinjen.

Ved å utstede et sertifikat til en web klientapplikasjon så vil en i prinsippet sertifisere brukeren som anvender denne klientapplikasjonen, i og med at det er brukerens identitet som inngår i sertifikatet. Det kan utstedes sertifikater til flere forskjellige brukere av en og samme klientapplikasjon, klientapplikasjonen må i så fall ha mekanismer for å beskytte nøkler anvendt av de forskjellige brukerne.

En klientapplikasjon som er sertifisert vil ha mulighet for å autentisere seg ovenfor kommunikasjonspartneren, typisk en web server. Hvis sertifikatet til klienten aksepteres, så vil klienten typisk få aksess til serveren og kan hente ned informasjon som ligger der. Dette tilsier at sertifikatet blir brukt som aksessertifikat og dermed kan en eliminere brukernavn, passord eller IP adresse som mekanismer for autentisering av klienten. I tillegg unngår en at informasjon som passord eller brukernavn går i klartekst over linjen i HTTP protokollen.

Referanser

[1] *UNInett SertifiseringsAutoritet (UNISA)*, Notat OMNI/01/98, O.E. Orøy, J.R. Sandbakken, J.Skretting, Norsk Regnesentral, Mars 1998

[2] *UNINETT Kryptotjeneste*, Notat DTEK/03/95, B.Hauksson, O.E.Orøy, S. Spurkland, J.Ølnes, Norsk Regnesentral, Juli 1995

[3] *FEIDE: Felles Elektronisk ID for UoH-sektoren*, Rapport no 963, ISBN 82-539-0467-3, Alf Hansen (UNINETT FAS), Anund Lie, Jon Ølnes, Norsk Regnesentral, September 2000

[4] *UNINETTs kryptotjeneste (UNISA)*, Foredrag UNINETT Konferansen 1995, Trondheim, O.E. Orøy, J.Ølnes, November 1995

[5] *Innføring og bruk av elektronisk legitimasjon*, Foredrag UNINETT konferansen 1997. Tromsø, O.E. Orøy, September 1997

[6] *Secure electronic mail - Authentication, encryption and certification*, Foredrag 17'th Nordic Internet Conference (NORDUnet'98). Tromsø, O.E. Orøy, June/July 1998

[7] *Kryptering uten kryptiske kommandoer*, O.E. Orøy, Foredrag Sikkerhetsseminaret i Bergen -- Universitetet i Bergen, Februar 1998

[8] *Sertifisering - Legitimasjon og infrastruktur*, Foredrag Uninett sikkerhets-kollokvium. Uninett, Trondheim, O.E. Orøy, Juni 1998

[9] *Sikker kommunikasjon - Hvordan anvende ditt sertifikat*, Foredrag Uninett sikkerhetskollokvium. Uninett, Trondheim, O.E. Orøy, Juni 1998

[10] *Sikkerhetsbegreper - Standarder, nytteverdi og politikk*, Foredrag Uninett sikkerhetskollokvium. Uninett, Trondheim, O.E. Orøy, Juni 1998

[11] *Heldagsseminar om Internett og sikkerhet*, Foredrag ENCATA prosjektseminar, Vestlandsforskning, Sogndal, J. Ølnes, O.E. Orøy, Mai 1997

[12] *Elektronisk legitimasjon du kan stole på*, Foredrag ICE-TEL avslutningsseminar. Norsk Regnesentral, O.E. Orøy, Desember 1997

[13] *Sertifikattjenester / PKI - status mm*, Foredrag UNINETT / USIT seminar om infratjenester, J.Ølnes, 21. februar 2000

[14] *Sikker utveksling av e-post*, Teknisk Ukeblad 2. november 1995 s.36, N.H. Berge, November 1995

[15] *UNINETT PCA policy statements*, Request for Commets (RFC) 1875, N.H. Berge 1995

[16] *Sertifisering av PGP nøkler*, UNINytt 1/97, J. Skretting og O.E. Orøy, UNINETT, 1997

[17] *ICE-CAR*, UNINytt 1/2-98, UNINETT 1998

[18] *Sikker overføring av data med kryptering i WWW og E-mail*, HØit 1/98, Trond Løvereide, Høgskolen i Østfold — avd. for Informatikk og Automatisering, 1998

[19] *Kan sikre din e-post*, Computerworld 6/98, s.10, E. Tønnessen, Computerworld 13.2.98

[20] *Final Text of Draft Amendment 1 to ISO/IEC 9594-8 on Certificate Extension*, ISO/IEC JTC 1/SC 21/WG 4 and ITU-T Q15/7, December 1996

[21] *Internet Public Key Infrastructure Part 1*, Russel Housley, Warwick Ford, Stephan Farell, David Solo, IETF PKIX Working Group July 1997, Internet Draft

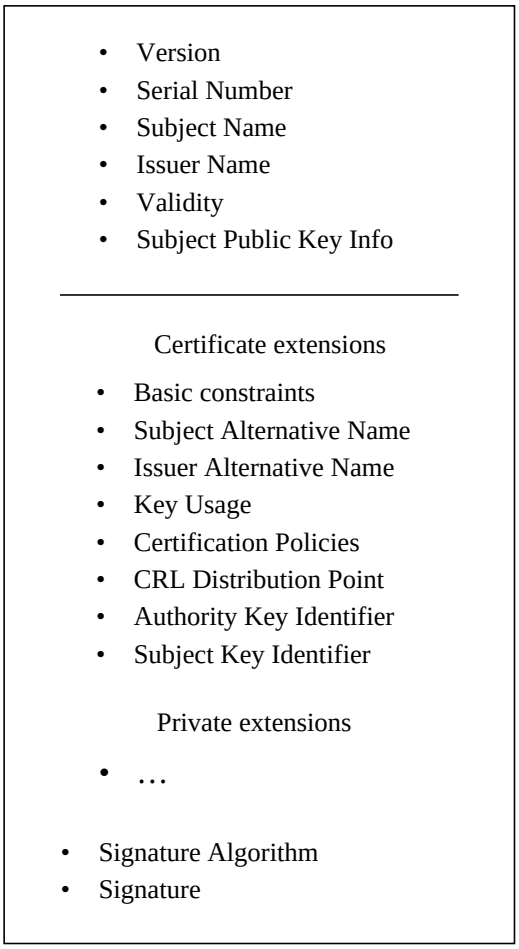
[22] *Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework*, RFC 2527, S.Chokhani, W.Ford, March 1999

[23] *Driftsdokumentasjon for UNISA tjenesten*, Internt UNISA notat, J.R. Sandbakken, Mai 1999

Vedlegg A X.509v3 Sertifikater

Dette vedlegget representerer en mindre oppdatering av Vedlegg A i [1].

X.509v3 sertifikatformatet er vist i Figur 3 og beskrevet nedenfor. Sertifikatet er en utvidelse av X.509v1 formatet. Se også [20] og [21] for mer detaljert informasjon.



Figur 3: X.509v3 sertifikat

Sertifikatet representerer først og fremst en binding mellom innehaveren og innehaverens offentlige nøkkel. Sertifikatet skal være autorisert av en utsteder og er ment å være offentlig tilgjengelig, eller i det minste tilgjengelig for de som har behov for å kommunisere sikkert med innehaveren. Det forutsettes at innehaveren beskytter og skjuler sin motsvarende hemmelige/private nøkkel for innsyn fra andre, og sertifikatet kan da benyttes til å:

- sende kryptert informasjon til innehaveren, med garanti om at bare han/hun kan dekryptere informasjonen
- kontrollere gyldighet av en digital signatur generert av innehaveren.

Utsteder av sertifikatet (CAen) garanterer for innholdet ved å signere på sertifikatet. Dette skjer ved at utsteder genererer en kryptografisk sjekksum av innholdet, krypterer denne verdien med sin hemmelige nøkkel og inkluderer resultatet (signaturen) i sertifikatet. Andre som har utsteders sertifikat kan validere sertifikatet ved å dekryptere signaturen (ved hjelp av utsteders offentlige nøkkel) og sammenligne sjekksummen med en egenberegnet sjekksum. Med dette får man validert både at utsteder har gått god for innholdet og at ingen siden har gjort endringer i sertifikatet.

Feltene Signature og Signature Algorithm nederst i figuren angir henholdsvis signaturen og algoritmen som benyttes i signeringen. SHA-1 for sjekksumberegning og RSA for kryptering er vanlig signeringsmetode.

X.509v3 er ellers et utvidbart format der ulike profiler kan tilpasses ulike behov og anvendelser. SECUDE programvaren støtter et sett av standardutvidelser, men ikke alle.

X.509v3 standardfelter

Standardfeltene, som er arvet fra v1-standarden, er:

- Version
  Angir X.509-versjon. Verdi 0 angir v1-sertifikater og verdi 2 angir v3-sertifikater.
- Subject Name
  Angir innehavers navn i form av et Distinguished Name (DN). Ved bruk av SECUDE er DN obligatorisk. X.509v3-standarden åpner dog får at man kan benytte alternative navn i utvidelsene.
- Issuer Name
  Angir tilsvarende utsteders DN.
- Validity
  Angir gyldighetsdatoene for sertifikatet.
- Subject Public Key Info
  Angir den offentlige nøkkelen og typen nøkkel, f.eks. en RSA nøkkel.

X.509v3 utvidelser

X.509v3 tilbyr et antall utvidelser, certificate extensions, i forhold til X.509v1. De som støttes av SECUDE programvaren er:

- Basic Constraints
  Angir hvorvidt eieren kan operere som CA eller ei. CA sertifikater skal ha verdi TRUE, mens brukere (og RAer) skal ha verdi FALSE i sertifikatet.
- Subject Alternative Name
  Han være av type:
  - RFC-822 Name,
  - URI,
  - Directory Name,
  - IP Address,

- Registered ID.

Førstnevnte angir mail-adressen til innehaveren, altså som et alternativt til DN-navnet. URI angir på sin side lokasjonen til sertifikatet med tanke på nedlasting. Protokollene *http, ftp, ldap* og *mailto* kan benyttes til dette. URIen kan også angi en sekvens av sertifikater, typisk sertifikater til CAer over i hierarkiet opp til roten. Merk altså at 'URI Name' ikke er et alternativt 'navn' på innehaveren f.eks. i form av en hjemmeside.

De tre øvrige navnealternativene benyttes ikke i UNISA.

Sertifikater kan inneholde flere 'Subject Alternative Names'.

- Issuer Alternative Name
  Angir tilsvarende data for utstederen.
- Key Usage
  Gir mulighet for å begrense anvendelsen av den offentlige nøkkelen på sertifikatet til en eller flere av de følgende alternativer:
  - Digital Signature
    For verifikasjon av signaturer som bare har rene autentiseringsformål.
  - Non Repudiation
    For verifikasjon av signaturer som har rene non repudiation formål.
  - Key Encipherment
    For kryptering av nøkler.
  - Data Encipherment
    For kryptering av andre data.

  For CAer kan man i tillegg angi:
  - Key Cert Sign
    For verifikasjon av sertifikatsignaturer.
  - CRL Sign
    For verifikasjon av signaturer av tilbakekallingslister (CRLer).

  Key Usage verdien er i UNISA sertifikater satt slik at sertifikatene kan benyttes i alle de ovennevte sammenhengene.
- Certificate Policies
  Angir policy informasjon i form av 'object identifiers' (OIDer). Disse refererer policier sertifikatet er utstedt under. *EuroPKI* roten har eksempelvis beskrevet sin sertifiseringspolicy i et dokument med Policy OID 1.3.6.1.4.1.5255.1.1.1, mens UNINETT PCA sin policy hadde OID 1.3.6.1.4.1.2428.10.1.2 da PCAen hadde egen policy. CAer må som minimum oppfylle policyen til overliggende CA, men kan innføre strengere policyer om ønskelig.
- CRL Distribution Point
  Angir lokasjonene til tilbakekallingslister (CRLer) ved hjelp av URIer for nedlasting. Støttede protokoller her er *http*, *ftp*, *ldap* og *mailto*.
- Authority Key Identifier
  Er en SHA-1-beregnet hash, dvs. en kryptografisk beregnet sjekksum, av utsteders offentlige nøkkel. Det er ellers mulig å ha flere offentlige nøkler på sertifikatene, selv om dette ikke benyttes i UNISA.
- Subject Key Identifier
  Er tilsvarende en SHA-1-beregnet hash av innehavers offentlige nøkkel.

Private utvidelser

X.509v3 formatet støtter videre private utvidelser, eller Private Extensions, som kommuniserende parter kan innføre gjerne uavhengig av andre. SECUDE programvaren støtter private utvidelser bl.a. med tanke på interoperabilitet med organisasjoner som Netscape, Microsoft og VeriSign. SECUDE vil ignorere ukjente, private utvidelser.

For å støtte bruk av S/MIME og sikker web-kommunikasjon (SSL) ved bruk av Netscape Navigator/Communicator eller Microsoft Internet Explorer, støtter man de private utvidelsene:

- Netscape Cert Type
- Netscape Comment

Flere Netscape extensions er definert, men disse benyttes ikke i UNISA. Cert Type er en 8 bits verdi som kan benyttes til å begrense anvendelsene av sertifikatet, hvilket kan være nyttig hvis man f.eks. opererer med flere sertifikater. De enkelte bit-ene angir (ved verdi 1) at sertifikatet er sertifisert til anvendelsene angitt i henhold til:

bit 0: for SSL-autentisering (for klienter)

bit 1: for SSL-autentisering (for servere)

bit 2: for bruk av S/MIME (for klienter)

bit 3: for signering av kodeobjekter (eksempelvis Java Applets)

bit 4: reservert

bit 5: for utstedelse SSL-sertifikater

bit 6: for utstedelse av S/MIME-sertifikater

bit 7: for utstedelse av sertifikater for objektkodesignering

I SECUDE spesifiseres Netscape Cert Type heksadesimalt og med bit 0 til venstre. Verdien ’80’ angir eksempelvis at bit 0 er satt.

Utvidelsen Netscape Comment er en tekstkommentar som kan vises i forbindelse med presentasjon av sertifikatet i Netscape.

Kritiske og ikke-kritiske utvidelser

Alle utvidelser i X.509v3 har et boolsk attributt som sier om den tilhørende utvidelsen er kritisk eller ikke (critical eller non-critical). I forbindelse med validering skal sertifikater med ukjente, kritiske utvidelser alltid forkastes. Ukjente, ikke-kritiske utvidelser kan ignoreres.

Av utvidelsene nevnt over som støttes av SECUDE, er Key Usage og Basic Constraint kritiske, mens de øvrige er ikke-kritiske. Det er eksempelvis da ikke mulig å presentere seg falskt som CA uten at dette oppdages i valideringen. Det er ellers opptil den enkelte CA å velge om Policy Identifier skal være kritisk, men dette er ikke tilfellet i UNISA.

Private utvidelser bør alltid være ikke-kritiske, slik at slike sertifikater kan benyttes også i sammenhenger der de private utvidelsene ikke er aktuelle eller ikke blir forstått.

Vedlegg B EuroPKI Certificate Policy

Der har vært flere versjoner av policier for UNISA PCA. RFC1875 [15] var den første. Deretter hadde UNINETT en med OID 1.3.6.1.4.1.2428.10.1.1[1] og siden en med OID 1.3.6.1.4.1.2428.10.1.2 i draft versjon, før det ble utarbeidet en felles policy for hele EuroPKI med OID 1.3.6.1.4.1.5255.1.1.1. EuroPKI policien er veldig mye basert på UNINETTs siste draft policy. Det er EuroPKI policien som er gjengitt i dette vedlegget og det er snakk om den versjonen som ble vedtatt 31.10.2000 på ICE-CAR møtet i Torino.

EuroPKI har formelt Copyright på policien, men det er likevel fritt fram å hente formuleringer fra den i henhold til avsnitt B-2.9.

I vedlegget her er policien gjengitt med layout som passer til dette dokumentet forøvrig.

1) UNINETT har som organisasjon fått tildelt OID 1.3.6.1.4.1.2428.10 og har så satt på et ett tall for "sikkerhetsfelt" og enda et ett tall for "den første UNISA/ICE policien".

B-1 Introduction

EuroPKI is a non-profit organization established to create and develop a pan-european public-key infrastructure (PKI). It has its roots in the PKI established by the ICE-TEL project and further developed by the ICE-CAR one. Both these projects were funded by the European Commission under the Telematics for Research programme.

More information is available at `http://www.europki.org/ca/root/`.

The structure of this document is according to RFC 2527 [1]. Therefore there are some sections that are maintained for compatibility, although they do not apply exactly to the services offered by EuroPKI. Appendix 1 provides a glossary of terms used in this document. It is mainly based on [1].

Within this document the words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", "OPTIONAL" are to be interpreted as in RFC 2119 [2]. (See Appendix 2.)

In this document the expression "conforming CA" is used to indicate a CA whose behaviour is conforming to the set of provisions specified in this document.

B-1.1 Overview

This document describes a set of rules that indicates the applicability of a certificate issued by conforming CA to its community of users and/or class of application with common security requirements.

A certificate policy MAY be used by a certificate user to help in deciding whether a certificate, and the binding therein, is sufficiently trustworthy for a particular application. An X.509 Version 3 certificate issued by a conforming CA SHOULD contain a reference to this certificate policy.

More detailed information about the practices which a conforming CA employs in its operations in issuing certificates can be found in the Certification Practice Statements (CPS).

Every conforming CA MUST issue its own CPS in order to provide information to potential clients of the CA about the underlying technical, procedural and legal foundations which are not specified in this policy.

B-1.2 Identification

This certificate policy is identified by the following unique registered Object Identifier (OID):

`1.3.6.1.4.1.5255.1.1.1`

The OID is composed by the following parts:

ISO assigned	1
Organization acknowledged by ISO	3
US Department of Defense	6

Internet	1
Private	4
IANA registered private enterprises	1
EuroPKI	5255
Root CA	1
Major version	1
Minor version	1

B-1.3 Community and applicability

A conforming CA can choose freely which are the community and applicability of their issued certificates but it MUST clearly specify them in its own CPS. In every case a conforming CA MUST NOT issue certificates to entities that don't belong to its community or for applications that haven't been carefully evaluated (for instance high value B2B transactions). Moreover a conforming CA SHALL respect all the limitations imposed by the following sections of this policy.

B-1.3.1 Certification Authority

An issuing conforming CA has to take particular care when it has to decide if a certain organization or individual can manage a subject CA performing all the controls and checks detailed in this policy.

A conforming CA MAY use as many RAs (registration authorities) as it wishes. A conforming CA MAY also have the role of RA if the entity authentication can be done by the CA itself. Subordinate CAs MUST sign an agreement with the certifying CA, stating the obligation to adhere to the agreed procedures.

B-1.3.2 Registration authorities

Registration Authorities (RA) are needed for physical identification/authentication of entities. These authorities MUST not be permitted to issue certificates.

A registration authority (RA) is

- an individualor
- a group of people appointed by an organization or an organizational unit

trusted by a CA, serving as a contact point for registration of new end entities, i.e. end entities that want to have a certificate issued. RAs have to check the certificates requester's identity in an appropriate way.

The RA MUST sign an agreement with the certifying CA, stating the obligation to adhere to the agreed procedures.

B-1.3.3 End entities

The end entities to be certified under this policy can be a natural person (individual or representing an organization) or a computer entity (e.g. a computer, a router or an application), capable of performing cryptographic operations.

Each conforming CA MUST detail in the CPS who are the end entities that it is willing to certify.

B-1.3.4 Applicability

One of the purposes of this policy is to promote a wide use of public-key certificates in many different applications. In order to promote interoperability this policy strongly encourages CA to support S/MIME for securing e-mail exchanges. It is also suggested that IPsec (to offer network layer security) and SSL/TLS (to offer transport layer security for protecting application protocols like HTTP, Telnet, FTP) SHOULD be supported. It's important to notice that this policy in principle doesn't want to put a priori limitation to the use of the certificates except for the case in which certificates are used in a way that is prohibited by the law of the countries where the issuing CA are established. However in order to evaluate if certificates issued under this policy are suitable for a certain application the chapter 2 about "General provisions" has to be read carefully and fully understood.

B-1.4 Contact Details

B-1.4.1 Specification administration organization

On behalf of EuroPKI this policy is fully managed by the Computer and Network Security Group (CNSG) of Politecnico di Torino, Italy (http://security.polito.it/).

B-1.4.2 1.4.2 Contact person

Contact point for questions related to this policy is:

address:
Prof. Antonio Lioy
EuroPKI Root Certification Authority
c/o Politecnico di Torino
Dip. Automatica e Informatica
corso Duca degli Abruzzi 24
10129 Torino (Italy)

phone: +39 0115647021 / +39 0115647054

fax: +39 0115647099

URI: http://www.europki.org/ca/root/

e-mail: ca@europki.org

B-1.4.3 Person determining CPS suitability for the policy

In order to obtain an evaluation of CPS suitability for the policy, conforming CAs have to contact the person mentioned in 1.4.2. See section 8.3 for details about CPS approval procedures.

B-2 General provisions

This chapter describes obligations for relevant parties and makes statements on liability, financial/economical issues. Moreover there's a section about confidentiality that classifies information into confidential information and publicly available and distributable information. Auditing statements are also located here.

B-2.1 Obligations

B-2.1.1 CA obligations

A conforming CA SHALL operate a certification authority service. The main obligations of a CA are:

- handle certificate requests and issue new certificates:
  - accept and confirm certification requests from entities requesting a certificate according to the agreed procedures contained in this policy and in the CPS;
  - authenticate entities requesting a certificate, possibly by the help of separately designated RAs;
  - issue certificates based on authenticated entities' requests;
  - send notification of issued certificate to requesters;
  - make issued certificates publicly available.
- handle certificate revocation requests and certificate revocation:
  - accept and confirm revocation requests from entities requesting a certificate to be revoked according to the agreed procedures contained in CPS/policy;
  - authenticate entities requesting a certificate to be revoked;
  - specify in the CPS what is the exact behaviour of the CA in issuing CRL (e.g. if the CA issues a CRL every time that a revocation occurs or if updated information will be available not before the time indicated in the NextUpadate field of the last issued CRL);
  - send notification of revocation of the certificate to requesters;
  - make CRLs publicly available.

B-2.1.2 RA obligations

An RA SHALL operate an RA service. This includes:

- to authenticate the identity of the subject;
- to validate the connection between a public key and the requester identity including a suitable proof of possession method;
- to confirm such validation versus the CA;
- to adhere to the agreement made with the CA.

B-2.1.3 Subscriber obligations

A subscriber SHALL behave according to the issuing CA CPS. This includes:

- to read and adhere to the agreed procedures;
- to properly protect its private key, being the only possessor if the subscription refers to an individual person. In the case of a private key of a hardware or software component the protection and the control of the key MAY be under the responsibilities of more than one authorized person;
- to accept that in the usage of public key certificates CA's liability is limited according to what is specified by section 2.2;
- to authorize the treatment and conservation of personal data;
- to notify immediately the CA upon private key compromise.

B-2.1.4 Relying party obligations

A relying party MUST be familiar with the CPS and this policy before drawing any conclusion on how much trust he can put in the use of a certificate issued from a conforming CA. A relying party MUST check CRLs when validating the use of a certificate. Moreover a relying party MUST ONLY use the certificate for the proscribed applications and MUST NOT use the certificates for forbidden applications.

B-2.1.5 Repository obligations

Each conforming CA SHALL use a publicly accessible repository to store certificates and Certificate Revocation Lists (CRLs).

The repository SHALL be available as much as practically possible.

B-2.2 Liability

B-2.2.1 CA liability

Conforming CA MAY accept liability. Considering that this policy is primarily established to promote the adoption of certificates as a mean to increase computer and network security in a broad variety of applications, the subsection 1.3.4 states that there are no a priori limitation to applicability of certificates issued under this policy. If no limitation is put on certificate applicability, this policy suggests that CA liability will be restricted to the guarantee of making the necessary controls to verify the identity of every requester as described in the CPS and to the adoption of the minimal security measures needed to protect CA's private key. In every case the complete list of accepted liabilities MUST be specified in the CPS.

B-2.2.2 RA liability

Cf. subsection 2.2.1.

B-2.3 Financial responsibility

With regards to what is stated in subsection 1.3.4, 2.2.1 and section 2.5, no financial responsibility is accepted for certificates issued under this certificate policy.

B-2.3.1 Indemnification by relying parties

No stipulation.

B-2.3.2 Fiduciary relationships

No stipulation.

B-2.3.3 Administrative processes

No stipulation.

B-2.4 Interpretation and Enforcement

B-2.4.1 Governing law

Interpretation of this policy is according to the law of the country where the conforming CA is established. This MUST be detailed in the CPS.

B-2.4.2 Severability, survival , merger, notice

No stipulation.

B-2.4.3 Dispute resolution procedures

No stipulation.

B-2.5 Fees

B-2.5.1 Certificate issuance or renewal fees

This policy suggests that no fees are charged for issuing certificates. However the CA MAY charge fees, but this MUST explicitly be stated in the CPS.

B-2.5.2 Certificate access fees

This policy suggests that no fees are charged for allowing certificate access. However the CA MAY charge fees, but this MUST explicitly be stated in the CPS.

B-2.5.3 Revocation or status information access fees

This policy suggests that no fees are charged for allowing certificates revocation or status information access.

B-2.5.4 Fees for other services such as policy information

This policy suggests that no fees are charged for allowing policy and CPS information access.

B-2.5.5 Refund policy

No stipulation.

B-2.6 Publication and Repository

B-2.6.1 Publication of CA information

A conforming CA SHALL make available:

- the policy and CPS it operates according to;
- all issued certificates except those certificates of subscribers that explicitly requested that their certificate SHALL not be made publicly available;
- signed certificate revocation lists.

B-2.6.2 Frequency of publication

Certificates SHALL be published as soon as they are issued. The frequency of CRL publication is specified in 4.4.9. Also policy and CPS SHALL be published as soon as they are updated.

B-2.6.3 Access control

There SHOULD be no access control to policy, CPS and CRL. There MAY be access control to certificates (for instance to prevent bulk acquisition of data like e-mail addresses or when CA decides to charge fees for certification services).

B-2.6.4 Repositories

There SHALL exist at least a repository for publishing the information mentioned above.

B-2.7 Compliance audit

No external audit is REQUIRED, only a self-assessment by the organization operating the conforming CA, that the operation is according to this policy. But any external compliance control is allowed. Conforming CA MAY verify that her subordinate CAs operate according this policy.

Every conforming CA MAY specify in the CPS more detailed information about compliance audit.

B-2.7.1 Frequency of entity compliance audit

No stipulation.

B-2.7.2 Identity/qualifications of auditor

No stipulation.

B-2.7.3 Auditor’s relationship to audited party

No stipulation.

B-2.7.4 Topics covered by audit

No stipulation.

B-2.7.5 Actions taken as a result of deficiency

No stipulation.

B-2.7.6 Communication of results

No stipulation.

B-2.8 Confidentiality

The CA collects personal information about the subscribers (e.g. full name, organization, and e-mail address). These data MUST be processed in a way that ensures privacy protection according to the laws of the country where the CA is established.

B-2.8.1 Types of information to be kept confidential

All subscribers' information that is not present in the certificate and CRL issued by a conforming CA is considered confidential and SHALL not be released outside if there is no explicit subscriber's authorization.

B-2.8.2 Types of information not considered confidential

Information included in public certificates and CRLs issued by a conforming CA are not considered confidential.

B-2.8.3 Disclosure of certificate revocation/suspension information

When a certificate is revoked/suspended, a reason code MAY be included in the CRL entry for the action. This reason code is not considered confidential and may be shared with all other users and relying parties. However, no other details concerning the revocation are normally disclosed.

B-2.8.4 Release to law enforcement officials

A conforming CA will not disclose certificate or certificate-related information to any third party, except when required by law enforcement officials that exhibit regular warrant.

B-2.8.5 Release as part of civil discovery

No stipulation.

B-2.8.6 Disclosure upon owner's request

A conforming CA will not disclose certificate or certificate-related information to any third party, except when required by the owner, with a signed request.

B-2.8.7 Other information release circumstances

No stipulation.

B-2.9 Intellectual Property Rights

A conforming CA MUST not claim any IPR on issued certificates.

Moreover anybody MAY copy from the EuroPKI CPS/policy, including a reference to the source.

B-3 Identification and authentication

This component describes the procedures used to identify and authenticate a certificate requester to a CA or RA before certificate issuance. It also describes how parties requesting rekey or revocation are authenticated. This component also addresses naming practices, including name ownership recognition and name dispute resolution.

B-3.1 Initial Registration

B-3.1.1 Types of names

The naming attributes of the subscriber to be requested to identify and authenticate the requester depend on the type of certificate that the subscriber requires.

In the choice of the types and format of names used in the fields of the certificate EuroPKI policy is conforming to RFC 2459 [3].

Conforming CA MUST detail in the CPS the types and format of names used.

B-3.1.2 Need for names to be meaningful

The Subject and Issuer name contained in a certificate MUST be meaningful in the sense that the issuing CA has proper evidence of the existent association between these names and the entities to which they belong.

If an e-mail address is included in the certificate this has not necessarily to follow a semantic rule that could be used to identify person and/or organization.

B-3.1.3 Rules for interpreting various name forms

Conforming CA MUST detail in the CPS the rules for interpreting various name forms used in the certificates.

B-3.1.4 Uniqueness of names

The DN MUST be unique for each subject entity certified by the one CA as defined by the issuer name field.

B-3.1.5 Name claim dispute resolution procedure

Disputes are managed according to the law of the country where the CA is established.

B-3.1.6 Recognition, authentication and role of trademarks

No stipulation.

B-3.1.7 Method to prove possession of private key

The adoption of proper method to prove possession of the private key corresponding to the public key being certified is strongly RECOMMENDED especially for signing key in order to provide non-repudiation of transactions.

The method adopted MUST be detailed in the CPS. If a method to prove possession is chosen, conforming CA MUST NOT issue certificate for which the proof of possession fails. This policy doesn't deem that the enforcement of a "key recovery" practice is a good proof of possession method. Indeed in this case CA has to figure out a mechanism to protect all the private keys collected.

B-3.1.8 Authentication of organization identity

Every time a subscriber requires the inclusion of the name of a certain organization in a certificate, issuing CA MUST must ensure that the subscriber is effectively related to the organization. In order to obtain this result issuing CA MUST require some documents. In all cases suitable legal documents that prove the data to be certified MUST be presented by means of out-of-band methods. The CA or RA MAY perform the authentication. The details MUST be specified in the CPS.

B-3.1.9 Authentication of individual identity

In many cases public-key certificates constitute a mean to guarantee strong cryptographic authentication of communicating entities. Bearing in mind this premise EuroPKI policy states that authentication of individual identity is REQUIRED. The RECOMMENDED method of authentication requires that individual presents personally to the authenticating CA or RA showing suitable identification documents (e.g. passport, driver's license, government badge etc.). Other methods like videoconference MAY be adopted. If the subject to be certified is a software component the person who submits the request MUST prove that he has the necessary authorization (As an example you can consider the request for the web server www.europki.org: only people directly authorized from who registered the domain europki.org can make such a request). The exact procedure MUST be detailed in the CPS.

B-3.2 Routine rekey

This policy doesn't mandate any compulsory rekey. After certificate expiration, the CA MAY issue a new certificate both for the same key or for a new key. The rekey authentication MAY be accomplished with the same procedure indicate in section 3.1 for initial registration or using digitally signed requests. These requests MUST be sent to the CA before certificate expiration.

A CA MAY issue more than one certificate for the same key.

B-3.3 Rekey after revocation

A public key whose certificate has been revoked for private key compromise MUST NOT be re-certified. The public key MAY be re-certified if the revocation is only due to certificate suspension. In the latter case the rekey authentication MAY be accomplished with the same procedure indicated in section 3.1 for initial registration or using digitally signed requests. These requests MUST be sent to the CA before certificate expiration.

B-3.4 Revocation request

A proper authentication method is required in order to accept revocation request. Conforming CA MUST accept as a revocation request a message digitally signed with a not expired and not previously revoked certificate issued under this policy. The same procedures adopted for the authentication during initial registration are also considered suitable. Alternative procedures MAY be supported such as secure communication of a revocation PIN (Personal Identification Number).

The exact procedures supported MUST be detailed in the CPS.

B-4 Operational requirements

B-4.1 Certificate Application

This policy permits two alternatives procedures for certificate application:

- certification of entities done entirely by the CA. The details about this procedure MUST be specified in the CPS;
- an entity generates its own key pair and submit public key and other required data to the CA. After that the request MUST carefully follow the procedures detailed in this policy and in the CPS for identification and authentication.

B-4.2 Certificate Issuance

Conforming CA and RA MUST carefully check the compliance and validity of documents presented by the subscribers. After the authentication accomplished by methods specified in section 3.1, CA SHOULD issue the certificate. In the case of issuance CA MUST notify the requester. If for any reasons CA decides not to issue the certificate (even if the checks and the authentication were correct) it SHOULD notify the reason for this choice to the requester.

B-4.3 Certificate Acceptance

No stipulation.

B-4.4 Certificate Suspension and Revocation

Conforming CA is responsible for issuing CRLs and for publishing signed versions. Although [3] doesn't require CAs to issue CRLs, conforming CA MUST issue timely CRLs.

The CA SHALL update its CRL with revoked subject CA certificates.

B-4.4.1 Circumstances for revocation

A certificate SHALL be revoked when information in the certificate is known to be suspected or compromised. This include situations where:

- the subscriber's data changed;
- the subscriber's private key is compromised or is suspected to have been compromised;
- the subscriber's information in the certificate is suspected to be inaccurate;
- the subscriber is known to have violated his obligations.

B-4.4.2 Who can request revocation

Conforming CA MUST accept a revocation request made by the holder of the certificate to be revoked. Moreover the revocation request MAY come from the CA that issued the certificate or from associated RA.

Other entities MAY require revocation, presenting evident proof of knowledge of the private key compromise or the change of subscriber's data.

B-4.4.3 Procedure for revocation request

The entity requesting the revocation SHALL be properly authenticated. The authentication method SHOULD be as strong as the one used in the issuing procedure. Conforming CA MUST accept as a revocation request a message digitally signed with a not expired and not previously revoked certificate issued under this policy. An alternative procedure MAY require the entity to visit RA or CA and to present a viable identity document.

If the entity is a CA, the CA SHALL in addition:

- Inform subscribers and cross-certifying CAs;
- Terminate the certificate and CRLs distribution service for certificates/CRLs issued using the compromised private key.

B-4.4.4 Revocation request grace period

The conforming CA decides what is the amount of time necessary to accept the request.

B-4.4.5 Circumstances for suspension

A CA MAY temporarily suspend a subscriber's certificate if the subscriber requests that service. Unlike revocation, suspension of a user allows for re-enabling at a later time. In every case conforming CA are not required to offer the suspension service.

Information on public keys of disabled users MAY be available from CA repository.

B-4.4.6 Who can request suspension

In the case that a CA offers the suspension service, CA MUST accept a suspension request made by the holder of the certificate to be suspended.

B-4.4.7 Procedure for suspension request

The entity requesting the suspension SHALL be properly authenticated. Conforming CA MUST accept as a suspension request a message digitally signed with a not expired and not previously revoked certificate issued under this policy. An alternative procedure MAY require the entity to visit RA or CA and to present a viable identity document.

B-4.4.8 Limits on suspension period

No stipulation.

B-4.4.9 CRL issuance frequency (if applicable)

CRLs MUST be issued at least every 40 days by conforming CA.

B-4.4.10 CRL checking requirements

Relying party MUST verify a certificate against the most recent CRL issued from conforming CA in order to validate the use of the certificate.

B-4.4.11 On-line revocation/status checking availability

Conforming CA MAY support on-line revocation/status checking. Bearing in mind that this policy requires conforming CA to issue CRL, it isn't mandatory to implement on-line revocation/status checking procedures. However this policy suggests taking into consideration OCSP [4] as such a mechanism.

B-4.4.12 On-line revocation checking requirements

No stipulation.

B-4.4.13 Other forms of revocation advertisements available

No stipulation.

B-4.4.14 Checking requirements for other forms of revocation advertisements

No stipulation.

B-4.5 Security Audit Procedures

This policy recognizes the importance of security audit procedures suggesting that conforming CA specifies all this kind of provisions in the CPS.

B-4.5.1 Types of event recorded

No stipulation.

B-4.5.2 Frequency of processing log

No stipulation.

B-4.5.3 Retention period for audit log

No stipulation.

B-4.5.4 Protection of audit log

No stipulation.

B-4.5.5 Audit log backup procedures

No stipulation.

B-4.5.6 Audit collection system (internal vs external)

No stipulation.

B-4.5.7 Notification to event-causing subject

No stipulation.

B-4.5.8 Vulnerability assessments

No stipulation.

B-4.6 Records Archival

This section specifies the type of events that are recorded for archival purposes from CA and RA and how this collected data are maintained. For further details not explicitly stipulated here the reference is the CPS.

B-4.6.1 Types of event recorded

Conforming CA SHOULD archive:

- certification requests corresponding to actually issued certificates;
- issued certificates;
- issued CRLs;
- all signed agreements with other parties (e.g. RA);
- document collected from the subscriber during the enrollment procedure;
- all relevant messages exchanged with RA.

The RAs SHOULD archive:

- all validation information collected from the subscriber;
- all relevant messages exchanged with CA.

B-4.6.2 Retention period for archive

The minimum retention period is 2 years.

B-4.6.3 Protection of archive

No stipulation.

B-4.6.4 Archive backup procedures

No stipulation.

B-4.6.5 Requirements for time-stamping of records

No stipulation.

B-4.6.6 Archive collection system (internal or external)

No stipulation.

B-4.6.7 Procedures to obtain and verify archive information

No stipulation.

B-4.7 Key changeover

No stipulation.

B-4.8 Compromise and Disaster Recovery

If a CA's private key is compromised or suspected to be compromised, the CA SHALL at least:

- inform subscribers, cross-certifying CAs and relying parties;
- terminate the certificates and CRLs distribution service for certificates/CRLs issued using the compromised private key;
- request the revocation of the CA's certificate.

If a RA's private key is compromised or suspected to be compromised, the RA SHALL at least inform the CA and request the revocation of the RA's certificate

If an entity's private key is compromised or suspected to be compromised, the entity SHALL at least inform the relying parties and request the revocation of the entity's certificate.

B-4.8.1 Computing resources, software, and/or data are corrupted

No stipulation.

B-4.8.2 Entity public key is revoked

No stipulation.

B-4.8.3 Entity key is compromised

No stipulation.

B-4.8.4 Secure facility after a natural or other type of disaster

No stipulation.

B-4.9 CA Termination

Termination of a CA is regarded as the situation where all service associated with a logical CA is terminated permanently.

Before the CA terminates its services the following procedures MUST be completed as a minimum:

- inform all subscribers, cross certifying CA's, higher level CAs, and relying parties with which the CA has agreements or other form of established relations;
- make publicly available information of its termination;
- stop distributing certificates and CRLs.

A subordinate CA MAY terminate or continue operation as a self-standing CA.

B-5 Physical, procedural, and personnel security controls

B-5.1 Physical Controls

Security requirements imposed on the conforming CA are indicated in the CPS. In every case this policy states that CA MUST be run on a dedicated workstation. The workstation MUST be physically secured.

B-5.1.1 Site locations and construction

No stipulation.

B-5.1.2 Physical access

The physical access to the site in which the CA operates MUST be restricted only to explicitly authorized people.

B-5.1.3 Power and air conditioning

No stipulation.

B-5.1.4 Water exposures

No stipulation.

B-5.1.5 Fire prevention and protection

No stipulation.

B-5.1.6 Media storage

No stipulation.

B-5.1.7 Waste disposal

No stipulation.

B-5.1.8 Off-site backup

No stipulation.

B-5.2 Procedural controls

All the issues related to procedural control like the definition of trusted roles MUST be specified in the CPS.

B-5.2.1 Trusted roles

No stipulation.

B-5.2.2 Number of person required per task

No stipulation.

B-5.2.3 Identification and authentication for each role

No stipulation.

B-5.3 Personnel controls

B-5.3.1 Background, qualifications, experience, and clearance requirements

The personnel operating the CA MUST be technically and professionally competent. Every conforming CA SHOULD specify in the CPS further details concerning this particular topic and the related issues.

B-5.3.2 Background check procedures

No stipulation.

B-5.3.3 Training requirements

No stipulation.

B-5.3.4 Retraining frequency and requirements

No stipulation.

B-5.3.5 Job rotation frequency and sequence

No stipulation.

B-5.3.6 Sanctions for unauthorized actions

No stipulation.

B-5.3.7 Contracting personnel requirements

No stipulation.

B-5.3.8 Documentation supplied to personnel

No stipulation.

B-6 Technical security controls

B-6.1 Key Pair Generation and Installation

This component is used to define the provisions for key management and the corresponding technical security controls.

B-6.1.1 Key pair generation

Conforming CA's cryptographic keys are generated by the package chosen for certificate handling.

End entities' cryptographic keys are locally generated by their application during the requesting process or by the CA during the enrollment procedure. This policy suggests the adoption of the former procedure for signing key pair to be used for non-repudiation purposes. The latter procedure MAY be adopted for encryption key pair or bulk authentication key pair.

B-6.1.2 Private key delivery to entity

The entity MAY generate his own key pair. It is important to notice that in the case of key pair generation done by CA, the key pair MUST be given to the end entity in a secure way. Further details MUST be specified in the CPS.

B-6.1.3 Public key delivery to certificate issuer

For individual certification, the entity SHALL submit a certification request containing the public key, locally generated, to the CA/RA. Every conforming CA MUST specify in its CPS the exact procedures for delivering public key.

For CAs' certification, the subject CA generates the key pair.

B-6.1.4 CA public key delivery to users

Conforming CA MUST provide mechanisms to deliver CA public key to the users in a trustworthy manner. Further details MUST be specified in the CPS.

In every case CA's public keys MUST be publicly available in a repository accessible via standard protocol such as HTTP or LDAP.

B-6.1.5 Key sizes

The minimum length of the private key of an end entity to be certified MUST be decided by the CA issuer and MUST NOT be less than the value of 512 bits. It is RECOMMENDED that the key would have a minimum length of 1024 bits.

A CA key pair MUST have a minimum length of 1024 bits. It is RECOMMENDED a length of 2048 bits.

B-6.1.6 Public key parameters generation

No stipulation.

B-6.1.7 Parameter quality checking

No stipulation.

B-6.1.8 Hardware/software key generation

The keys can be generated in software or in hardware (e.g. on a cryptodevice) depending on the various tools available to the entities.

B-6.1.9 Key usage purposes (as per X.509 v3 key usage field)

The purposes for which a key can be used MAY be restricted by a CA through the KeyUsage extension in the certificate. This is a field that indicates the purpose for which the certified public key is used.

Certificates issued under this policy MUST have the KeyUsage extension flagged as critical. This means that the certificate SHALL be used only for a purpose for which the corresponding key usage bit is set to one.

CA Certificates

In CA's certificates KeyUsage extension MUST contain the following bits set to one:

```
digitalSignature - nonRepudiation - keyCertSign - cRLSign
```

It MAY contain also other bits set to one.

B-6.2 Private Key Protection

B-6.2.1 Standards for cryptographic module

This policy doesn't mandate the adoption of cryptographic module compliant with predetermined standards. Every conforming CA MAY give in the CPS more details about the adoption of standard compliant module.

B-6.2.2 Private key (n out of m) multi-person control

The private key of individual MUST NOT be under (n out of m) multi-person control. Only private keys belonging to a CA, a hardware component or a software component MAY be under such a control: in this case the type of control MUST be specified in the CPS.

B-6.2.3 Private key escrow

This policy discourages the implementation of private key escrow policy both for end entities and CA. Implementation of such policies MAY be permitted if and only if the governing law of the country in which the CA is established explicitly requires them.

B-6.2.4 Private key backup

This policy suggests that all the parties SHOULD maintain a backup copy of the private key in order to reconstitute it in case of destruction of the key. This backup MUST be carefully protected especially in the case of backup of private key CA.

B-6.2.5 Private key archival

This policy suggests the implementation of a procedure for private key archival only for private key used for encryption/decryption. Indeed it MAY be necessary to maintain a copy of a private key in order to correctly decrypt messages even if the corresponding public-key certificate is expired.

B-6.2.6 Private key entry into cryptographic module

The private key of all entities SHOULD be stored in an encrypted form. This provision is particularly important if the entity is a CA.

B-6.2.7 Method of activating private key

Specific details about how to activate private key SHOULD be found in the CPS. As a general suggestion this policy recommends that for the activation of a private key some specific activation data MUST be entered in the cryptographic module. At least the activation data MUST consist in a PIN or passphrase, but for the most valuable private key (e.g. the ones belonging to CA) the use of hardware tokens or biometrics data is suggested.

B-6.2.8 Method of deactivating private key

No stipulation.

B-6.2.9 Method of destroying private key

No stipulation.

B-6.3 Other aspects of key pair management

B-6.3.1 Public key archival

Conforming CA MUST archive all issued certificates. Mechanisms to provide integrity controls other than digital signatures MAY be implemented.

B-6.3.2 Usage periods for the public and private keys

No stipulation.

B-6.4 Activation data

B-6.4.1 Activation data generation and installation

Pass phrases or PINs SHALL be selected according to "best practice". This means that it is necessary to suggest a suitable minimal length for the pass phrases and to enforce mechanisms to check that pass phrases show enough entropy.

B-6.4.2 Activation data protection

Pass phrases protecting private keys SHALL be accessible only to the legitimate users (e.g. certificate holder for personal certificates, CA operators for CA signing keys, etc). An exception for this indication is the implementation of a secure archival/backup mechanism for activation data. Such a mechanism MUST be clearly defined in the CPS.

B-6.4.3 Other aspects of activation data

No stipulation.

B-6.5 Computer security controls

B-6.5.1 Specific computer security technical requirements

No stipulation.

B-6.5.2 Computer security rating

No stipulation.

B-6.6 Life cycle technical controls

B-6.6.1 System development controls

No stipulation.

B-6.6.2 Security management controls

No stipulation.

B-6.6.3 Life cycle security rating

No stipulation.

B-6.7 Network security controls

This policy strongly suggests that the machine on which the cryptographic module used for CA operations SHOULD be kept off-line to prevent network attacks. In every case network access to the CA workstation MUST be limited in order to protect the CA's private key in an appropriate way from disclosure.

B-6.8 Cryptographic module engineering controls

No stipulation.

B-7 Certificate and CRL profiles

B-7.1 Certificate Profile

In order to promote interoperability this policy strongly encourages conforming CA to issue certificates profiling them accordingly to [3]. In every case CPS MUST detail the specific profile adopted.

B-7.1.1 Version number(s)

The version field in the certificate SHALL state 2, indicating X.509v3 certificates.

B-7.1.2 Certificate extensions

In compliance with [3], the inclusion of the following certificate extensions is RECOMMENDED:

Extension name	
SubjectKeyIdentifier AuthorityKeyIdentifier	NOT CRITICAL NOT CRITICAL
BasicConstraints	CRITICAL
KeyUsage	CRITICAL
CertificatePolicies	NOT CRITICAL

It is also RECOMMENDED the use of other two extensions: CRLDistributionPoint for providing information useful to retrieve the CRL, and SubjectAltNames when there is the need to include an RFC822 e-mail address to a certificate. Both these two extensions SHOULD be marked as NOT CRITICAL.

B-7.1.3 Algorithm object identifiers

No stipulation.

B-7.1.4 Name forms

All related issue MUST be specified in the CPS.

B-7.1.5 Name constraints

All related issue MUST be specified in the CPS.

B-7.1.6 Certificate policy Object Identifier

Other certificate policy object identifiers are applicable if and only if the other policies identified are compliant with this policy. Conforming CA MUST contact the maintainers of the various policies to verify the level of mutual compliance. However in order to promote interoperability, following RFC 2459, this policy suggests to include only one certificate policy object identifier in a certificate.

B-7.1.7 Usage of policy constrains extension

All related issue MUST be specified in the CPS.

B-7.1.8 Policy qualifiers syntax and semantics

The Certificate Policies extension field has a provision for conveying, along with each certificate policy identifier, additional policy-dependent information in a qualifier field.

This policy suggests that the qualifier field SHOULD be a CPS Pointer qualifier that contains a pointer to a Certification Practice Statement (CPS) published by the CA.

The pointer is in the form of a uniform resource identifier (URI).

B-7.2 CRL Profile

B-7.2.1 Version number(s)

The version field in the certificate SHALL state 1, indicating X.509v2 CRL.

B-7.2.2 CRL and CRL entry extensions

No stipulation.

B-8 Specification administration

B-8.1 Specification change procedures

Editorial changes can be made to the policy and CPS. In case of substantial changes of the policy all CAs and users SHALL be notified in advance. Moreover CAs SHALL update the policy in accordance with the policy changes.

Policy changes that imply minor technical adjustments SHALL be notified in advance.

B-8.2 Publication and notification policies

This policy is available via Web at the URI

```
http://www.europki.org/ca/root/cps/
```

B-8.3 CPS approval procedures

Conforming CA MUST be evaluated for compliance with this policy. In order to obtain CPS approval conforming CAs MAY submit their CPS to the contact people specified in section 1.4.3. After that conforming CA MUST wait for the answer. The time limit for completing the evaluation is established in 60 days. It might be acceptable to have CA self certification for compliance, but in this case if non-compliance is reported to EuroPKI organization then the CA certificate will be revoked.

Policy Appendix 1

Certification Authority (CA) - An authority trusted by one or more users to create and assign public key certificates. Optionally the CA may create the user's keys. It is important to note that the CA is responsible for the public key certificates during their whole lifetime, not just for issuing them.

CA-certificate - A certificate for one CA's public key issued by another CA.

Certificate policy (CP) - A named set of rules that indicates the applicability of a certificate to a particular community and/or class of application with common security requirements. For example, a particular certificate policy might indicate applicability of a type of certificate to the authentication of electronic data interchange transactions for the trading of goods within a given price range.

Certification path - An ordered sequence of certificates which, together with the public key of the initial object in the path, can be processed to obtain that of the final object in the path.

Certification Practice Statement (CPS) - A statement of the practices which a certification authority employs in issuing certificates.

Certificate revocation list (CRL) - A CRL is a time stamped list identifying revoked certificates which is signed by a CA and made freely available in a public repository.

Issuing certification authority (issuing CA) - In the context of a particular certificate, the issuing CA is the CA that issued the certificate (see also Subject certification authority).

Public Key Certificate (PKC) - A data structure containing the public key of an end entity and some other information, which is digitally signed with the private key of the CA which issued it.

Public Key Infrastructure (PKI) - The set of hardware, software, people, policies and procedures needed to create, manage, store, distribute, and revoke PKCs based on public-key cryptography.

Registration authority (RA) - An entity that is responsible for identification and authentication of certificate subjects, but that does not sign or issue certificates (i.e., an RA is delegated certain tasks on behalf of a CA). [Note: The term Local Registration Authority (LRA) is used elsewhere for the same concept.]

Relying party - A recipient of a certificate who acts in reliance on that certificate and/or digital signatures verified using that certificate. In this document, the terms "certificate user" and "relying party" are used interchangeably.

Subject certification authority (subject CA) - In the context of a particular CA-certificate, the subject CA is the CA whose public key is certified in the certificate.

IPR - Intellectual Property Rights.

Policy Appendix 2: Key words for use in RFCs to Indicate Requirement Levels

According to RFC 2119 [2] "Key words for use in RFCs to Indicate Requirement Levels", we specify how the main keywords used in RFCs should be interpreted.

Authors who follow these guidelines should incorporate this phrase near the beginning of their document:

> The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHAL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119.

1. **MUST** This word, or the terms "REQUIRED" or "SHALL", mean that the definition is an absolute requirement of the specification.
2. **MUST NOT** This phrase, or the phrase "SHALL NOT", mean that the definition is an absolute prohibition of the specification.
3. **SHOULD** This word, or the adjective "RECOMMENDED", mean that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.
4. **SHOULD NOT** This phrase, or the phrase "NOT RECOMMENDED" mean that there may exist valid reasons in particular circumstances when the particular behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.
5. **MAY** This word, or the adjective "OPTIONAL", mean that an item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because the vendor feels that it enhances the product while another vendor may omit the same item. An implementation which does not include a particular option MUST be prepared to interoperate with another implementation which does include the option, though perhaps with reduced functionality. In the same vein an implementation which does include a particular option MUST be prepared to interoperate with another implementation which does not include the option (except, of course, for the feature the option provides).

Policy References

[1] RFC 2527 "Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework" March 1999
ftp://ftp.isi.edu/in-notes/rfc2527.txt

[2] RFC 2119 "Key words for use in RFCs to Indicate Requirement Levels" March 1997
ftp://ftp.isi.edu/in-notes/rfc2119.txt

[3] RFC 2459 "Internet X.509 Public Key Infrastructure: Certificate and CRL Profile" January 1999
ftp://ftp.isi.edu/in-notes/rfc2459.txt

[4] RFC 2560 "Internet X.509 Public Key Infrastructure: Online Certificate Status Protocol - OCSP" June 1999
ftp://ftp.isi.edu/in-notes/rfc2560.txt

[5] UNISA Certificate Policy
http://www.uninett.no/pca/index-e.html

[6] ICE-TEL policy
http://ice-tel.uni-c.dk/ice-ca/?h=3

Vedlegg C UNISA Certification Practice Statements

De practice statements som er gjengitt her er de praktiske forordningene som UNISA har fulgt med hensyn på å implementere EuroPKI policien.

Policien som er gjengitt i Vedlegg B er i skrivende stund av så ny dato at tilhørende practice statements fra EuroPKI fremdeles ikke er ferdig utarbeidet. Slike practice statements vil være på samme format som policien og de vil fortelle hvilke praktiske forordninger EuroPKI selv vil etterleve for å oppfylle policien. UNISA skulle så ha utarbeidet tilsvarende practice statements minst like kraftige som EuroPKI sine egne. Men UNISA prosjektet avsluttes før EuroPKI sine practice statements er på plass så de pratice statements som gjengis her er de UNISA har brukt i pilotperioden.

C-1 Certification of Persons

This document provides information about the certification practice with respect to the certification of persons undertaken by CAs (Certification Authorities) certified by the UNINETT PKI service UNISA.

The purpose is to provide information to members of the Internet community who wish to evaluate the trust they can place in a certificate issued to a person by a CA certified under the UNISA Certificate Policy.

The UNISA certificate policy identification is:

Certificate policy name:	**UNISA Certificate Policy**
Policy identifier:	`1.3.6.1.4.1.2428.10.1.2`
Distinguished Name (DN):	`OU=pca, O=uninett, C=no`
Subject alternative names:	RFC822: `pca@uninett.no`
	URI: `http://www.uninett.no/pca/`

Main contact point for questions related to these Certification Practice Statements (CPSs) is: `krypto-hjelp@uninett.no`.

Persons as stated in the title of this document is hereafter denoted **entities**.

C-1.1 Certification Procedure

The entity to be certified can be an end entity, an RA or a CA. In the case of an RA or a CA, certain agreements will have to be signed with the certifying CA as prescribed in the UNISA Certificate Policy.

C-1.1.1 Certification Contact Point

Information about the certification contact point is available on the UNISA information pages, which is at the following web location: *http://www.uninett.no/pca/*.

According to the policy, both individual certification and bulk certification can take place. In the case of individual certification, a web page exists which describes how to proceed to have a certificate issued.

C-1.1.2 Key Generation and Certification Request

In the case of individual certification, the entity must first generate his own key pair and self-signed certificate, and then send the self-signed certificate to the CA by e-mail.

See *Entity Naming Conventions* in C-1.3 for a description of what names are required as identity in the certificate.

The certification request must contain a key with minimum length of 512 bits, as shorter keys will not be certified by the CA.

The self-signed certificate sent to the CA can be in the form of a PEM message, S/MIME message, and as a PKCS #7, PKCS #10, X.509 Certificate, Netscape's SignedPublicKeyAndChallenge and possibly other. The certificate eventually returned from the CA will be returned as a PEM message, PKCS #7 ContentInfo or X.509 Certificate.

If the entity requests re-certification of the same public key, or can sign the certification request with any key belonging to that entity with at least the same key length and certified under this policy and not expired nor revoked, then all that is required is for the entity to send a signed certification request to the CA. The entity will in that case not receive a certification receipt (as described in C-1.1.3) nor is any further identity validation (as described in C-1.1.4) required.

In the case of bulk certification, the CA will generate both keys and certification requests.

A private key shall always be accessible only through an encrypted file storage, or through use of a smartcard authorised by UNISA. The key storage shall be protected by use of a pass phrase.

C-1.1.3 Certification Receipt

In the case of individual certification, the entity will receive an e-mail receipt from the CA, indicating that the certificate request is received by the CA.

C-1.1.4 Entity Identity Validation

In the case of individual certification, the entity's identity information that goes into the self-signed certificate must be verified before the CA can issue the certificate. The entity must therefore, according to the UNISA policy, visit the organization's RA (Registration Authority) (or the CA) to show proof of identity.

The entity must provide name and self-signed certificate fingerprint to the RA. The RA will verify the entity's identity by:

- driver's licence
- passport
- bank card (Norwegian)

The RA will also verify that the entity belongs to the set of entities that the CA recognizes as belonging to the RA's organization.

If personal identity and affiliation are both verified, the RA will sign over the entity's name and self-signed certificate fingerprint, and forward the signed message to the CA.

In the case of bulk certification, the entities' affiliation will be settled prior to certificate issuance. After certificate issuance, the entity can collect the pass phrase to activate the private key at the RA, given proof of identity. Alternatively, the pass phrase can be sent by certified surface mail.

In the absence of RAs, the identity of the entity must be verified by "out of band" means. These means will vary from case to case, depending on physical distance, prior knowledge etc.

CA and RA need not be separate entities. A CA can act in the role of an RA in addition to the CA role.

C-1.1.5 Certification Notification

In the case of individual certification, the CA will verify the entity's certification request and the name and fingerprint received from the RA, and issue a certificate for the entity if

the verification is successful. If the request is for re-certification, then the CA will issue a certificate if only the signature can be verified according to C-1.1.2. The entity will receive the issued certificate from the CA by e-mail.

In the case of bulk certification, an entity's key pair together with the certificate will either be DES-encrypted or put on a smartcard, and then forwarded to the entity. The pass phrase to generate the DES-key or the smartcard pin code is sent to the RA or by certified surface mail.

C-1.1.6 Certificate Retrieval and Installation

The entity must install the received certificate.

C-1.1.7 Entity Certificate Verification

The entity is recommended to verify the received certificate and verify the name of the issuing authority. In addition, it is also recommended to install locally the issuing authority certificate as described in C-1.2.

C-1.2 Authority Certificate Retrieval

The entity should install locally the certificate of the CA, and the certificates of all superior CAs including the root-CA. Information on how to retrieve CA certificates is available on the UNISA information pages.

A web page will be available to the entity describing how to proceed to retrieve a CA certificate.

C-1.3 Entity Naming Conventions

The entity identity is given by a Distinguished Name (DN), reflecting the organization to which the entity is affiliated and his full name, by the entity's e-mail address and its URI. The use of a DN is required, while e-mail address and URI are optional. It is strongly recommended to use both DN and e-mail address for identification.

Entities' DNs will follow the conventions described in the UNISA policy. It is the certifying CA who will ultimately determine an entity's DN, and the uniqueness of a DN.

C-1.4 Certificate Revocation

Certificate Revocation Lists (CRLs) will be issued at least once a month by CAs. Each CA will include a CRL distribution point extension as a URI in issued certificates. This URI gives the location where the CRLs can be found. Details on how CRLs can be found is described on the UNISA information pages.

It is the responsibility of the relying party to verify a certificate against the CRLs that the CA issues. As there is no automatic distribution service for CRLs to the relying parties, a relying party is itself responsible for retrieving and verifying CRLs.

CRLs can be retrieved by accessing the URI given in the CRL distribution point extension in the entity certificate.

If an entity:

- knows that the private key is compromised,
- have suspicion that the private key is compromised, or
- suspects the information in the certificate to be inaccurate,

the entity shall notify the CA that issued the certificate so that the certificate can be revoked. A revocation request should preferably be submitted to the CA in a message signed with the certificate to be revoked. Second best is for the entity to bring along some piece of ID and visit the RA who will forward a revocation request to the CA. The final method is to notify the CA in other ways.

If the entity is a CA, the CA shall in addition:

- inform subscribers and cross certifying CAs;
- terminate the certificate and CRL distribution service for certificates/CRLs issued using the compromised private key.

An entity's certificate shall also be revoked when the private key of the entity's RA is (suspected to be) compromised or in the case that the RA makes significant mistakes.

C-2 Certification of WWW Client applications

This document provides information about the certification practice with respect to the certification of WWW client applications undertaken by CAs (Certification Authorities) certified by the UNINETT PKI service UNISA.

Its purpose is to provide information to members of the Internet community who wish to evaluate the trust they can place in a certificate issued to a WWW client application by a CA certified under the UNISA Certificate Policy.

The UNISA certificate policy identification is:

Certificate policy name:	**UNISA Certificate Policy**
Policy identifier:	`1.3.6.1.4.1.2428.10.1.2`
Distinguished Name (DN):	`OU=pca, O=uninett, C=no`
Subject alternative names:	RFC822: `pca@uninett.no`
	URI: `http://www.uninett.no/pca/`

Main contact point for questions related to these Certification Practice Statements (CPSs) is: `krypto-hjelp@uninett.no`.

The person operating a **WWW Client application** as stated in the title of this document, is hereafter denoted an **entity**. By **client application** is meant the same as **WWW Client application**.

C-2.1 Certification Procedure

The entity to be certified can be an RA. If so, then an agreement will have to be signed with the certifying CA as prescribed in the UNISA Certificate Policy.

C-2.1.1 Certification Contact Point

Information about the certification contact point is available on the UNISA information pages, which is at the following web location: *http://www.uninett.no/pca/*.

Only individual certification is possible for WWW Client applications and a web page is available to entities describing how to proceed to have a certificate issued.

C-2.1.2 Key Generation and Certification Request

On the web page the entity must first enter the necessary identification information about himself, which will to be put in the certificate. The client application will then activate the generation of its own key pair and its own self-signed certificate, and submit the self-signed certificate to the CA by e-mail.

See *Entity Naming Conventions* in C-2.3 for a description of what names are required.

The certification request must contain a key with minimum key length of 512 bits, as shorter keys will not be certified by a CA.

If the entity requests re-certification of the same public key, or can sign the certification request with any key belonging to that entity with at least the same key length and certified under this policy and not expired nor revoked, then all that is required is to send a signed certification request to the CA. The entity will in that case not receive a certification receipt

(as described in C-2.1.3) nor is any further identity validation (as described in C-2.1.4) required.

Private keys shall be accessible only through an encrypted file storage, or through use of smartcards authorised by UNISA. The key storage shall be protected by use of a pass phrase.

C-2.1.3 Certification Receipt

The entity will immediately after submitting the self-signed certificate receive a receipt., indicating that the certificate request is forwarded to the CA.

C-2.1.4 Entity Identity Validation

The entity identity information that goes into the self-signed certificate must be verified before the CA can issue the certificate. The entity must therefore, according to the UNISA policy, visit the RA, or the CA, appropriate to his organization to show proof of identity.

The entity must provide its name and self-signed certificate fingerprint to the RA. The RA will verify the entity's identity by:

- driver's licence
- passport
- bank card (Norwegian)

The RA will also verify that the entity belongs to the set of entities that the CA recognizes as belonging to the RA's organization.

If personal identity and affiliation are both verified, the RA will sign over the entity's name and self-signed certificate fingerprint, and forward the signed message to the CA.

In the absence of RAs, the identity of the entity must be verified by "out of band" means. These means will vary from case to case, depending on physical distance, prior knowledge etc.

CA and RA need not be separate entities. A CA can act in the role of an RA in addition to the CA role.

C-2.1.5 Certification Notification

The CA will verify the entity's certificate request and the name and fingerprint received from the RA, and issue a certificate for the entity if the verification is successful. If the request is for re-certification, then the CA will issue a certificate if only the signature can be verified according to C-2.1.2.

The entity will receive a notification by e-mail, indicating that the certificate has been issued, and the web/ldap location where the certificate can be retrieved by the client application.

C-2.1.6 Certificate Retrieval and Installation

The entity must access the web /ldap location where the certificate is made available, and the client application will install the certificate.

C-2.1.7 Entity Certificate Verification

The entity is recommended to view the retrieved certificate and verify the name of the issuing authority. In addition, the entity is also recommended to download the issuing authority certificate as described in C-2.2. This will allow for verification of the entity's certificate signature.

C-2.2 Authority Certificate Retrieval

The entity should download the certificate of its CA, and the certificates of all superior CAs including the root-CA. Information on how to retrieve CA certificates is available on the UNISA information pages.

A web page will be available to the entity describing how to proceed to retrieve a CA certificate.

C-2.3 Entity Naming Conventions

The entity name shall be the name of the person operating the WWW client application, even if the key pair only can be used in the context of the client application. The identity that goes into the certificate reflects the person, and not the identity of the client application that is used by the person.

The entity's identity is given by a Distinguished Name (DN), reflecting the organization to which he belongs and the full name, and by the entity's e-mail address or ots URI. The use of a DN is required, while e-mail address and URI are optional.

Entities' DNs will follow the conventions described in the UNISA policy. It is the certifying CA who will ultimately determine an entity's DN, and the uniqueness of a DN.

C-2.4 Certificate Revocation

Certificate Revocation Lists (CRLs) will be issued at least once a month by CAs. Each CA will include a CRL distribution point extension as a URI in issued certificates. These URIs gives the location where the CRLs can be found. Details on how CRLs can be found is described on the UNISA information pages.

It is the responsibility of the relying party to verify a certificate against the CRLs that the CA issues. As there is no automatic distribution service for CRLs to the relying parties, a relying party is itself responsible for retrieving and verifying CRLs.

CRLs can be retrieved by accessing the URI given in the CRL distribution point extension in the entity certificate.

If an entity:

- knows that the private key is compromised,
- have suspicion that the private key is compromised, or
- suspects the information in the certificate to be inaccurate,

the entity shall notify the CA that issued the certificate so that the certificate can be revoked. A revocation request should preferably be submitted to the CA in a message signed with the certificate to be revoked. Second best is for the entity to bring along some piece

of ID and visit the RA who will forward a revocation request to the CA. The final method is to notify the CA in other ways.

An entity's certificate shall also be revoked when the private key of the entity's RA is (suspected to be) compromised or in the case that the RA makes significant mistakes.

C-3 Certification of WWW Server applications

This document provides information about the certification practice with respect to the certification of WWW server applications undertaken by CAs (Certification Authorities) certified by the UNINETT PKI service UNISA.

The purpose is to provide information to members of the Internet community who wish to evaluate the trust they can place in a certificate issued to a WWW server application by a CA certified under the UNISA Certificate Policy.

The UNISA certificate policy identification is:

Certificate policy name:	**UNISA Certificate Policy**
Policy identifier:	`1.3.6.1.4.1.2428.10.1.2`
Distinguished Name (DN):	`OU=pca, O=uninett, C=no`
Subject alternative names:	RFC822: `pca@uninett.no`
	URI: `http://www.uninett.no/pca/`

Main contact point for questions related to these Certification Practice Statements (CPSs) is: `krypto-hjelp@uninett.no`.

C-3.1 Certification Procedure

C-3.1.1 Certification Contact Point

Information about the certification contact point is available on the UNISA information pages, which is at the following web location: *http://www.uninett.no/pca/*.

A web page is available to the server administrator which describes how to proceed to have a certificate issued.

C-3.1.2 Key Generation and Certification Request

The server administrator must first have the server application generate its own key pair and self-signed certificate, and then enter the self-signed certificate on the provided web page for submission to the CA.

See *Server Naming Conventions* in C-3.3 for a description of what names are required as identity in the server application certificate.

The certification request must contain a key with minimum key length of 512 bits, as shorter keys will not be certified by a CA.

If the server administrator requests re-certification of the same public key, or can sign the certification request with any key belonging to that administrator with at least the same key length and certified under this policy and not expired nor revoked, then all that is required is to send a signed certification request to the CA. The server administrator will in that case not receive a certification receipt (as described in C-3.1.3) nor is any further identity validation (as described in C-3.1.4) required.

Private keys shall be accessible only through an encrypted file storage, or through use of smartcards authorised by UNISA. The key storage shall be protected by use of a pass phrase.

C-3.1.3 Certification Receipt

The server administrator will immediately after submitting the self-signed certificate receive a receipt, indicating that the certificate request is forwarded to the CA.

C-3.1.4 Server Identity Validation

The server identity information that goes into the self-signed certificate must be verified before the CA can issue the certificate. The server administrator must therefore, according to the UNISA policy, visit the RA, or the CA, appropriate to his organization to show proof of identity.

The server administrator must provide its name and self-signed certificate fingerprint to the RA. The RA will verify the server administrator's identity by:

- driver's licence
- passport
- bank card (Norwegian)

The RA will also verify that the server administrator belongs to the set of entities that the CA recognizes as belonging to the RA's organization. In addition, the RA will verify the privileges of the server administrator that enables him to have the server certificate issued.

If personal identity, affiliation and privileges are all verified, the RA will sign over the server administrator's name and self-signed certificate fingerprint, and forward the signed message to the CA.

In the absence of RAs, the identity of the server administrator must be verified by "out of band" means. These means will vary from case to case, depending on physical distance, prior knowledge etc. However, the organization that owns the server application must sign a written agreement as presented by the CA, to have the CA issue the certificate in the absence of an RA.

CA and RA need not be separate entities. A CA can act in the role of an RA in addition to the CA role.

C-3.1.5 Certification Notification

The CA will verify the certificate request, and issue a certificate for the server if the verification is successful. The server administrator will receive the issued certificate from the CA by e-mail.

C-3.1.6 Certificate Retrieval and Installation

The server administrator must install the certificate.

C-3.1.7 Server Certificate Verification

The server administrator is recommended to view the received certificate and verify the name of the issuing authority. In addition, it is also recommended to download the issuing authority certificate as described in C-3.2. This will allow for verification of the server's certificate signature.

C-3.2 Authority Certificate Retrieval

The server administrator should download the certificate of the CA, and the certificates of all superior CAs including the root-CA. Information on how to retrieve CA certificates is available on the UNISA information pages.

A web page will be available to the server administrator describing how to proceed to retrieve a CA certificate.

C-3.3 Server Naming Conventions

The server name shall be the name of the WWW server application. The identity that goes into the certificate reflects the server itself.

The server identity is given by a Distinguished Name (DN), reflecting the organization to which the server belongs, and the server's name (e.g. *www.uninett.no*), and by an e-mail address (e.g. *webmaster@uninett.no*). The use of both a DN and e-mail address is required.

Server DNs will follow the conventions described in the UNISA policy. It is the certifying CA who will ultimately determine a server's DN, and the uniqueness of a DN.

C-3.4 Certificate Revocation

Certificate Revocation Lists (CRLs) will be issued at least once a month by CAs. Each CA will include a CRL distribution point extension as a URI in issued certificates. These URIs gives the location where the CRLs can be found. Details on how CRLs can be found is described on the UNISA information pages.

It is the responsibility of the relying party to verify a certificate against the CRLs that the CA issues. As there is no automatic distribution service for CRLs to the relying parties, a relying party is itself responsible for retrieving and verifying CRLs.

CRLs can be retrieved by accessing the URI given in the CRL distribution point extension in the user certificate.

If a server administrator:

- knows that the private key is compromised,
- have suspicion that the private key is compromised,
- suspects the information in the certificate to be inaccurate, or
- loses server administrator privileges,

the server administrator shall notify the CA that issued the certificate, so that the certificate can be revoked. A revocation request can be submitted to the CA in a message signed with a certificate issued by UNISA, belonging to the server administrator and not expired nor revoked. A second possibility is for the server administrator to bring along some piece of ID and visit the RA who will forward a revocation request to the CA. The final method is to notify the CA in other ways.

Vedlegg D CA og RA avtaler

De CA og RA avtalene som har vært inngått i prosjektet følger malene som er gjengitt i dette vedlegget.

Drift av Sertifiseringsautoritet (CA)

Denne avtalen gjøres gjeldende mellom UNINETT PCA og ..... (heretter benevnt CA) om utstedelse av sertifikat for CA.

1. Identifikasjon og informasjon om CA

CAs DN: OU=_____, O=_____, C=_____
(OU trenger ikke å inngå i DN-et.)

Følgende personer vil stå for drift av CAen (maksimum 4 personer):
1. _____
2. _____
3. _____
4. _____
Disse personene er kjent med innholdet i denne kontrakten.

CA-sertifikatet vil være gyldig f.o.m. utstedelsesdatoen, og for en periode på ____ år.

Tilbakekallingslister (CRLer) skal oppdateres og offentliggjøres minst en gang i måneden. Offentliggjørelse av CRLer gjøres ved å legge CRLen inn i X.500/LDAP.

2. CAs plikter

CA skal:
- Håndtere sin sertifiseringsautoritet i henhold til den Policy og de Practice Statements for sertifisering av personer, web-tjenere og web-klienter som til enhver tid gjelder. Gjeldende Policy og Practice Statements på avtaletidspunktet er gjengitt i avtalens vedlegg 1 til 4.
- Tilby sine brukere låsesmedtjenesten (se info fra UNISA web).
- Holde UNINETT PCA oppdatert på endringer i de dataene som inngår i del 1.
- Umiddelbart informere UNINETT PCA om alle forhold som kan ha betydning for sikkerheten i signaturer utstedt med CAs private nøkkel.

3. UNINETT PCAs plikter

UNINETT PCA skal:
- Utstede et sertifikat for CA, etter å ha foretatt tilstrekkelig kontroll av identiteten til de aktuelle personene.
- Dersom PCA varsles om forhold som kan bety at CAs private nøkkel kan være på avveie, utstede en Certificate Revocation List (CRL) inneholdene CAens sertifikat, og publisere dette via X.500/LDAP (dvs. tilbakekalle sertifikatet).
- Dersom PCA varsles om at CA ikke opptrer i henhold til denne avtalen, kontrollere dette med CA, og eventuelt tilbakekalle sertifikatet.
- Sørge for at nytt sertifikat for CA utstedes i god tid før det forrige sertifikatet er utløpt.

For CA:	**For UNINETT PCA:**
Dato:	Dato:
Navn/stilling:	Navn/stilling:
Underskrift:	Underskrift:

Avtale om drift av Sertifiseringsautoritet (CA)

Denne avtalen inngåes mellom UNINETT Sertifiseringsautoritet (UNISA) og

_____

(heretter benevnt avtaletaker) om drift av CA på vegne av avtaletaker.

Generelle vilkår

CA skal opprettes med følgende Distinguished Name (DN):

_____

Avtaletaker kan når som helst si opp avtalen, og selv overta driftsansvar for sertifiseringsautoriteteten (CAen).

UNISA skal:

- drive en sertifiseringsautoritet (CA) på vegne av avtaletaker fra dags dato til 31.12.2000.
- drive sertifiseringsautoriteten i henhold til de retningslinjer som er beskrevet i UNISA Certificate Policy.

Avtaletaker skal:

- utnevne minst en person som kan fungere som lokal registreringsautoritet (RA), og som kan gi veiledning til organisasjonens brukere.

Avtaletaker delegerer herved ansvar for drift av sertifiseringsautoritet (CA) for sitt domene til UNISA.

For UNISA	For avtaletaker
_____	_____
Dato	Dato
_____	_____

Avtale om å opptre som registreringsautoritet (RA) på vegne av tilhørende sertifiseringsautoritet (CA)

Avtalen består av denne underskrevene siden, samt Practice Statements for sertifisering av personer, web-tjenere og web-klienter som er gjengitt i vedleggene 1 til 3.

1. CAs identitet

Organisasjon:_____

Distinguished Name (DN):_____

2. RAs identitet

Personnavn:_____

Distinguished Name (DN):_____

3. RAs plikter

- Håndtere sin registreringsautoritet i henhold til de Practice Statements for sertifisering av personer, web-tjenere og web-klienter som til enhver tid gjelder. Gjeldende Practice
  Statements på avtaletidspunktet er gjengitt i avtalens vedlegg 1 til 3.
- Å varsle CA så fort som mulig dersom det er sannsynlig at nøkkelen kan ha blitt kompromittert.

4. CAs plikter

- Å utstede sertifikat for RA som kan brukes til autentisering av brukere.
- Holde RA informert om hvilke brukere som kan sertifiseres.
- Tilbakekalle RAs sertifikat ved mistanke om at nøkkelen kan være kompromittert.

RA:	**For CA**:
Dato:	Dato:
Navn / stilling:	Navn / stilling:
Underskrift:	Underskrift:

Avtale med Sertifiseringsautoritet (CA)

Denne avtalen gjøres gjeldende mellom UNINETT PGP Policy Certification Authority

og ..... (heretter benevnt CA) om signering av CAs offentlige nøkkel. Avtalen består av denne siden, samt vedlegg 1 "Sertifiseringsstrategi".

Med 'sertifiseringsautoritet' forstås i denne avtalen en instans som sertifiserer brukere i henhold til en sikkerhetsstrategi. UNINETT PGP Policy Certification Authority er en slik sertifiseringsautoritet, som drives av UNINETT A/S (foretaksnummer 968100211) i henhold til "UNINETT PGP PCA Policy Statements".

1. Identifikasjon og informasjon om CA

CAs e-post adresse: _____

CAs navn (etternavn, fornavn): _____

CAs organisasjon: _____

2. CAs plikter

CA skal:

- Holde UNINETT PGP PCA oppdatert på endringer i de dataene som inngår i del 1 ovenfor.
- Signere UNINETT PGP PCA sin offentlige nøkkel
- Håndtere sin sertifiseringsautoritet i henhold til vedlegg 1 og "UNINETT PGP PCA Policy Statements".
- Utstede et "key compromise certificate" og sende dette til UNINETT PGP PCA, hvis CA eller UNINETT PGP PCA mener at CAs hemmelige nøkkel er kompromittert.
- Kontakte UNINETT PGP PCA hvis CA ønsker å fratre sin rolle som sertifiseringsautoritet.

3. UNINETT PGP PCAs plikter

UNINETT PGP PCA skal:

- Signere CA sin offentlige nøkkel, etter å ha foretatt tilstrekkelig kontroll av identiteten til CA.
- Publisere på sine web-sider hvilke CAer som er sertifisert av UNINETT PGP PCA.
- Publisere på sine web-sider hvilke CAer som eventuelt ikke lenger er å stole på.
- Publisere på sine web-sider eventuelle "key compromise certificates".
- Dersom UNINETT PGP PCA varsles om at CA ikke opptrer i henhold til denne avtalen,
  så skal UNINETT PGP CA først søke å avklare dette forholdet med CA, deretter vil UNINETT PGP PCA eventuelt publisere på sin web side at CA ikke lenger oppfyller sine plikter.

CA	**For UNINETT PGP PCA**
Dato:	Dato:
Navn:	Navn:
Underskrift:	Underskrift:

Sertifiseringsstrategi

Vedlegg 1

1. Sikkerhetsstrategi

Sikkerhetsstrategien er beskrevet “UNINETT PGP PCA Policy Statements”. CA plikter å følge denne strategien.

2. Navngivning

Enkeltpersoner som skal sertifiseres av CA skal bruke sin e-post adresse for unik identifikasjon, og skal angi identitet på følgende måte:

Navn [, organisasjonsnavn] <e-post adresse>

Bruk av organisasjonsnavn er valgfritt. Normalt vil navn være brukerens fulle navn. Det er
sertifiserende CA sitt ansvar å påse at navn og e-post adressen er korrekt, slik at det ikke
oppstår en navnekonflikt.

3. Gangen ved sertifisering av en brukere

1. Brukeren genererer en kopi av sin offentlige nøkkel, og sender denne til CA.
2. Brukeren oppsøker deretter sin CA for å bevise sin identitet, med mindre den offentlige nøkkelen ble sendt som en signert PEM melding som kan valideres av UNISA. Som gyldig legitimasjon regnes pass, førerkort og bankkort. Hvis den offentlige nøkkelen ble sendt som en signert PEM melding validerbar av UNISA, vil det alene være tilstrekkelig som
   identifikasjon.
3. CA må validere brukerens nøkkel ved å sjekke fingerprint, navn og e-post adresse. Hvis alt er i orden skal CA signere brukerens nøkkel, og returnere den signerte nøkkelen til
   brukeren.
4. Brukeren bør så signere CAens offentlige nøkkel utifra reint praktiske hensyn.

4. Krav til sikkerhet

- CAs nøkkel skal være minimum 1024 bit lang.
- Nøkkelen skal kun kunne aksesseres fra en DES/IDEA-kryptert fil/katalog, eller UNINETT PGP PCA godkjent smartkort, beskyttet av et passord.
- Nøkkelen eller passordet skal under ingen omstendigheter lagres i klartekst noe sted.
- Ingen andre enn CA skal kunne passordet.
- Det forventes at “skikk og bruk” ved valg av passord følges.
- Passordet skal bestå av flere ord (også kalt pass phrase).

5. Informasjon til brukere

UNINETT PGP PCA skal opprettholde en informasjonstjeneste for sine brukere og

CAer vedrørende sertifisering, eller andre spørsmål knyttet til bruk av sikkerhetstjenesten. Dette i form av en e-post adresse og en webside. UNINETT PGP PCA skal så langt det lar seg gjøre forsøke å besvare spørsmål fra CAene.

UNINETT PGP PCA vil publisere navn og fingerprint for alle sertifiserte CAer på sin web side. UNINETT PGP PCA forbeholder seg retten til å publisere navn på alle CAer som har fått sin nøkkel kompromittert, eller som ikke oppfyller de kravene som settes til en CA.

Vedlegg E PGP Policy

Dette vedlegget inneholder den policien som ble brukt ifm. PGP tjenesten da den var aktiv. Policien er datert Juni 1997, har tittel *UNINETT PGP PCA Policy Statements* og ble skrevet som en RFC (selv om det aldri var aktuelt å gi den ut som en slik) av kategori *informational* med følgende introduksjon:

> This memo provides information for the Internet community. Although it is written in the form and style of an RFC, it is not intended for publication as an RFC. Distribution of this memo is unlimited.

I vedlegget her er policien gjengitt med layout som passer her og ikke med RFC layout. Innholdet er dog det samme.

Introduction

This document provides information about policy statements submitted by the UNINETT Pretty Good Privacy Policy Certification Authority (UNINETT PGP PCA).

Its purpose is to provide information to members of the Internet community who wish to evaluate the trust they can place in a certification path that includes a certificate issued by the UNINETT PGP PCA, or to set up a CA to be certified by the UNINETT PCA.

PGP PCA Identity

The Uninett PGP PCA will be identified by the name:

```
"Uninett PGP Policy Certification Authority <pgp-ca@uninett.no>"
```

The email address for the PGP PCA will be:

```
pgp-ca@uninett.no
```

The UNINETT PGP PCA will be run by:

```
Norwegian Computing Centre
Gaustadallien 23
P.O.Box 114 Blindern,
N-0314 Oslo, Norway
```

Contact person:

```
Odd Egil Orøy
Email: Odd.Egil.Oroey@nr.no
```

```
Tel.: (+47) 22 85 25 00
Fax : (+47) 22 69 76 60
```

Duration: This policy is valid from June 1. 1997 to Jan. 1. 2000

Info about the PGP PCA is available at: `http://www.uninett.no/pca/pgp.html`

UNINETT - a Brief Overview

UNINETT is a Limited Company (AS) operating the Norwegian network for academics and research. It is incorporated under Norwegian law, and it's company number is 968100211.

More information is available from the UNINETT web server at: `http://www.uninett.no/`

PGP PCA Scope and Purpose

UNINETT already runs a PCA for certification of CAs (Certification Authority) issuing X.509 certificates, this service is named UNISA. The primary aim of the UNISA service is to build an infrastructure for PEM (Privacy Enhanced Mail)[1-4], although the certificates may be used for any purpose, for instance S/MIME. The PCA policy is documented in RFC 1875 [5].

The use of PGP (Pretty Good Privacy) [6] for protected transfer and storage of information has become widespread in the Internet community. The UNINETT PGP PCA shall act as a top level (root) CA within the UNINETT domain, realising a common point of trust for those PGP users within this domain that choose to be certified by CAs certified by the PGP PCA.

Additionally the PGP PCA may facilitate establishment of trust paths between a PGP user within the UNINETT constituency and a PGP user outside of this constituency, by certification of the UNINETT PGP PCA against similar services in other constituencies. Where the certification must obey this policy.

The UNINETT PGP PCA service is an offer to PGP users within the domain. PGP users must deliberately choose to get certified by CAs certified by the PGP PCA, and no restrictions are implied on the ability of PGP users to sign the certificates of one another.

PGP's trust model is the "web of trust". This is designed to work without any infrastructure, by having the users themselves sign certificates for one another.

Experience shows that non-hierarchical structures, like a web of trust, have problems related to scaling. For the web of trust, the main problems are the length of chains of trust, and determining the degree of confidence one can have in the users that are involved in this chain.

The idea behind the UNINETT PGP PCA is to utilise the infrastructure already established by UNINETT to offer a hierarchical trust model to PGP users. The confidence of PGP

certificates issued under this hierarchy is determined by the certification policy defined in this document.

The hierarchy consists of only two levels: the PGP PCA and CAs. This limits the length of the trust paths within the hierarchy. PGP users certified by CAs may in turn sign certificates of other PGP users, thus creating further levels, or shortcuts, in the hierarchy. Such certification is not within the scope of this policy or of the UNINETT PGP PCA.

The CAs certified by the PGP PCA are certified as persons. It is assumed that a CA will predominantly work within its organisation, and the prime targets of the PGP PCA service are organisations within the UNINETT domain. However there is no restriction on a CA's right to sign certificates for persons outside its organisation, nor is the PGP PCA itself restricted to certify CAs only within the UNINETT domain. The only restriction is that the PGP PCA policy must be obeyed.

PGP PCA Security & Privacy

Security Requirements Imposed on the PGP PCA

- The PGP PCA will run on a dedicated workstation with no network connection. The workstation shall be physically secured.
- The PGP PCA will use a standard PGP implementation with keys stored on the disk of the workstation and all cryptographic processing in software. Use of a smartcard based implementation is a future option.
- Exchanging data between the PGP PCA and the rest of the world will be done by use of tapes or floppy discs.
- The PCA RSA key pair will have a length of 1024 bits.
- Backups from the PCA workstation must be stored in at least one off site location. Backups will be physically secured.

Security Requirements Imposed on CAs

- The person certified as CA may perform his/her CA tasks by use of the same PGP installation that he/she uses for PGP communication. This shall be a standard PGP implementation.
- The CA's private key shall be accessible only through a IDEA/DES encrypted file storage, or through use of smartcards authorised by UNINETT PGP PCA. The key storage shall be protected by use of a pass phrase selected according to "best practice".
- The PGP installation shall run on a workstation which may be connected to a network. The workstation must, however, be adequatly secured from attacks originating from an open network by applying "common sense" protection mechanisms.
- The PGP installation shall be stored on a local disk connected to the workstation. If a smartcard is used, the card reader must be local to the workstation.
- Backups must be physically secured.
- The CA RSA key pair must have a minimum length of 1024 bits.

Measures taken to protect the privacy of any information collected in the course of certifying CAs and (for CAs) users.

Neither the PCA nor CAs will collect any security relevant information about users. Users and CAs will always generate their own key pairs. The only information stored are the CA certificates at the PGP PCA, which are considered as non-confidential information.

Certification Policy

Policy and Procedures when Certifying CAs

In order to be certified, a CA must sign an agreement with the UNINETT PGP PCA stating the obligation to adhere to the agreed procedures.

The person responsible for running the CA will be evaluated by the UNINETT PCA, in order to determine whether he/she exhibits the necessary qualifications and have access to the resources needed in order to run the CA securely.

The PGP PCA is responsible for sufficient validation of the identity of the person responsible for the CA.

The CA must submit a copy of its public key to the UNINETT PGP PCA. The PGP PCA must check the name, email address and fingerprint associated with the key, before signing a PGP certificate for the CA. A UNISA validated PEM message incorporating the public key is sufficient for validation.

The CA shall subsequently sign the key of the UNINETT PGP PCA after sufficient validation of the correctness of this key.

Policy and Procedures when Certifying Users

A user will generate his/her own key pair. A copy of the public key shall be sent to the CA by email.

The user must then visit the CA carrying a proof of his/her identity, or by sending the public key as a UNISA validated PEM message, in that case further identification is unnecessary.

A person’s identity is verified by:

- driver’s licence
- passport
- bank card (Norwegian)

The CA validates the user’s public key by checking the fingerprint of the key, the user’s name and email address. The CA then signs the user’s key, and returns the result to the user.

The user is then recommended to sign the CA’s public key, following normal PGP procedures.

Naming Conventions

Email addresses are used as unique identification of users. Identities are specified in PGP certificates as:

```
Name [, Organisation] <email address>
```

The name should be the real name of the user. The Organisation part is optional. The CA must ensure that name and email address are correct and that no naming conflicts will occure.

Certificate Management

Key Distribution

UNINETT PGP PCA will maintain a list of names and key fingerprints for all authorised CAs. This will be published through a web page. UNINETT PGP PCA reserves the right to publish names of CAs whose keys have been compromised, or who otherwise do not anymore fulfil the requirements for the CA role.

UNINETT will maintain a web page and an email enquiry service for information to users. All web pages for the service will be placed under `http://www.uninett.no/pca/pgp.html`

No service for distribution of the users' PGP keys is offered. It is referred to existing key archives on the Internet.

Key Revocation

Certificate Revocation Lists are not used with PGP. If a CA's private key is compromised, the CA must issue a "key compromise certificate". UNINETT PGP PCA will distribute this through its web pages, but the CA shall also ensure distribution of this information. Uninett PGP PCA reserve the right to publish the name of the CAs that are no longer considered trusted, even if a "key compromise certificate" has not been issued.

All users certified by this CA should request new certificates.

Procedures for revocation of users' keys are not covered by this policy.

Distribution of Software

UNINETT PGP PCA will not distribute PGP software. It is referred to software archives on the Internet. Note that European PGP users must fetch the software from a non-US source.

Security Considerations

Security issues are discussed throughout this memo.

References

[1] Linn, J., "Privacy Enhancement for Internet Electronic Mail: Part I: Message Encryption and Authentication Procedures", RFC 1421, IAB IRTF PSRG, IETF PEM WG, February 1993.

[2] Kent, S., "Privacy Enhancement for Internet Electronic Mail: Part II: Certificate-Based Key Management", RFC 1422, IAB IRTF PSRG, IETF PEM, BBN, February 1993.

[3] Balenson, D., "Privacy Enhancement for Internet Electronic Mail: Part III: Algorithms, Modes, and Identifiers", RFC 1423, IAB IRTF PSRG, IETF PEM WG, TIS, February 1993.

[4] Kaliski, B., "Privacy Enhancement for Internet Electronic Mail: Part IV: Key Certification and Related Services", RFC 1424, RSA Laboratories, February 1993.

[5] Berge, N., "UNINETT PCA Policy Statements", RFC 1875, December 1995.

[6] Zimmermann, P., "PGP Users Guide Volume I", October 1994.

[7] Schneier, B., "E-mail security", John Wiley & Sons, Inc, 1995.

Authors' Address

```
Jon Ølnes, Odd Egil Orøy and Jonn Skretting
Norwegian Computing Centre
Gaustadalleen 23
P.O.Box 114 Blindern,
N-0314 Oslo, Norway

Phone: (+47) 22 85 25 00
Fax :  (+47) 22 69 76 60
EMail: Jon.Olnes@nr.no, Odd.Egil.Oroey@nr.no, Jonn.Skretting@nr.no
```

Vedlegg F Driftsdokumentasjon

Denne driftsdokumentasjonen er en oppdatering av driftsdokumentasjonen i [23]. Den aller vesentliste endringen er relatert til beskrivelsen av distribusjonsmekansimer for sertifikater og tilbakekallingslister, hvor LDAP er introdusert etter at [23] ble forfattet.

NR har kun benyttet SECUDE kommandogrensesnitt på Unix og det er det som er dokumentert her. Det ser ut som at det nå også er kommet grafisk grensenitt for CA og RA verktøy på NT platform, men det har vi ikke prøvd ut.

F-1 Installasjon

Installasjon av SECUDE programvare

SECUDE programvaren kan i skrivende stund hentes via web fra GMD Darmstadt eller fra SECUDE. Aktuelle URLer er:

- `http://www.darmstadt.gmd.de/secude/`
- `http://www.secude.com/`

Filnavnet vil normalt avspeile versjon, operativsystem og noe om hvorvidt den er statisk linket eller ei, som f.eks. `sec5.1c-sunos-stat.tar.Z`.

Programvaren kan pakkes ut ved hjelp av:

```
# cat <secude-program>.tar.Z | uncompress | tar xvf -
```

Filene kan godt eies av `root`, men det ser ut til at det fungerer like godt med en uprivilegert eier. Om man som gruppeeier ønsker å ha f.eks. `uninett` så kan man gjøre en:

```
# chgrp -R uninett <secude-katalog>
```

Installasjon av smartkortleser

Det følgende angir installasjon av SNI smartkortleser under SECUDE versjon 5.2b.

Smartkortleseren koples til serieporten på maskinen. Deretter utføres:

```
secude> sctinst
  No SCT's configured
sctinst> new
  sct_id   ? 1
  port     ? /dev/ttya
  reade    ? snisct
  parms    ? <RETURN>
  comment  ? <RETURN>
sctinst> <RETURN>
  Save file (yes/no)? y
```

Smartkorleseren testet først uten smartkort i:

```
secude> sctest -b 1
  ------ Evaluate SCT with sct_id 1 ( 'sctest -b 1' )
  Information about SC PSE
  SCT type: Siemens Nixdorf B1 Reader on port /dev/ttya
```

og deretter med smartkort i leseren:

```
secude> sctest -b 1
  ------ Evaluate SCT with sct_id 1 ( 'sctest -b 1' )
  Information about SC PSE
  SC interface: SECUDE SmartCard Plugin 2.00 for TCOS Card
  SCT type : Siemens Nixdorf B1 Reader on port /dev/ttya
  SC info  : TCOS Card V2.0 (using CRT) with
             application
  SC serial: <serienr>
  SC alg   : RSA with these key sizes: 1024
```

Installasjon av LDAP server

Vi har benyttet Enterprise Directory Server (EDS) fra Messaging Direct (Isode Ltd). Den er lisensbelagt, men UNINETT har (hatt) lisens.

Generelt kan man si at EDS er meget tung å installere om man skal kompilere den selv, der var ingen `configure` mekanismer innebygget. Men det følger med mye dokumentasjon for EDS som sådan og den er grei å lese. Det finnes også HTML til LDAP/X.500 gateway og en Directory Data Manager (DDM[1]) som kan benyttes for å manipulere data i katalogen. Vi har imidlertid benyttet *tcldish*, som også kommer med, som klient.

Programvare hentes fra:

- `http://www.isode.com/cust_prepage.html`

Brukernavn og passord må oppgis. Informasjon om hvordan utpakking skjer finnes på:

- `http://www.isode.com/customer_only/source/index.html`

Man må også hente ned en *Customer Keyfile*, som anvendes sammen med utpakkingsprogrammene og dedikerte brukernavn/passord for å pakke ut pakkene med kildekode. Informasjon omkring dette finnes på:

- `http://www.isode.com/source/C-4509.html`

Når man har installert programvaren så må man generere en faktisk katalog. Til dette bruker man `edm`. Denne startes ved:

```
% sbin/edm
```

og den har et grafisk grensesnitt for håndtering (management) av LDAP katalogen. For å lage en ny katalog velg *File->New->Directory Server...* UNISAs katalog har hatt navn `cn=X509v3 Directory, o=Uninett, c=NO`. Editer *Presentation Adress* ved å markere for LDAP og oppgi server host og port. Når katalogen er generert så kan den startes vhja. `edm` ved å velge *Server->Start a server* i menyen til `edm`. Og `edm` kan brukes til å stoppe serveren. For å administrere katalogen mer detaljert så velger man *File->Open...* Man må da huske det passordet man oppga da katalogen ble generert. Og så er det bare å lese manualene…

Konfigurasjonsfiler

etc/oidtable.oc

Denne editeres (om nødvendig) som følger:

- flytt alle mandatories til optional for `certificationAuthority` og legg til `mail`:

  kommer som:

```
certificationAuthority:standardObjectClass.16 : top : \
cACertificate, certificateRevocationList, \
authorityRevocationList: crossCertificatePair
```

1) Det så ikke ut som om at UNINETT hadde lisens, men vi kunne fått den gjennom ICE siden Messaging Direct var med der, om vi så hadde ønsket.

la bli til:

```
certificationAuthority:standardObjectClass.16 : top : : \
cACertificate, certificateRevocationList, \
authorityRevocationList, crossCertificatePair, mail
```

etc/oidtable.at

- Denne må editeres om nødvendig for å kunne dytte inn SECUDE sertifikater, følgende endringer må gjøres:
  - finn alle attributter med syntaks `certificate` og endre til `ASN`, slik som:

    ```
    userCertificate:attributeType.36 :Certificate
    cACertificate:attributeType.37 :Certificate
    serverCertificate: ic-ms-at.19 :Certificate
    mtsCertificate: ic-ms-at.20 :Certificate
    mtaCertificate: ic-ms-at.21 :Certificate
    msCertificate: ic-ms-at.22 :Certificate
    userSMIMECertificate: netscape-at.40 :Certificate
    ```

- Det er ikke sikkert at *ldif* understøttes

etc/ldaptailor

Her kan du sette opp hvilken DSA som skal kalles initielt, f.eks.:
dsa_address x509 LDAP://unisa.nr.no:1522/

~/.duarc

Denne fila leses ved start av *tcldish* og kan f.eks. se slik ut:

```
username: <cn=DSA Manager,cn=X509v3 Directory,o=UNINETT,c=NO>
bind: -simple
service: -sizelimit 15
list: -sizelimit 30
```

~/.dishdraft

Denne fila leses av *tcldish* kommandoen `dadd` dersom det ikke er gitt noen `-f` opsjon. Innholdet kommer opp i din favoritt editor for editering. I det tilfellet fila ikke eksisterer så vil den bli initialisert med en hel rekke attributter. Jmf. ellers F-9.

F-2 Generering av PCAer og CAer

Grunnleggende oppsett

For PCAen og for hver CA opprettes det en egen Unix bruker som eier. Denne administrerer og opererer så autoriteten. Brukernavnene reflekterer autoritetens navn. Brukernavnene for dagens CAer, samt tilhørende hjemmekatalog mm., er angitt i Tabell 6.

Brukerne har hatt `tcshell` som oppstart-shell og følgende må være satt i de respektives `.cshrc-fil` på hjemmekatalogen:

```
setenv SECUDE_ETC $HOME/etc
setenv LD_LIBRARY_PATH /local/packages/<secude-katalog>/lib
setenv PATH /local/packages/<secude-katalog>/bin:$PATH
```

I tillegg skal det opprettes en katalog `secude` på `$HOME`. Sjekk oppsettet vhja.:

```
secude> info
```

Det har vært vanlig å legge alle autoritetbrukere i en felles Unix-gruppe kalt `uninett`.

De opprettede Unix brukerne kan så starte SECUDE programvaren vhja. kommandoen:

```
% secude
```

En oversikt over kommandoer i SECUDE får man vhja. kommandoen:

```
secude> help
```

Før man begynner å generere en PCA/CA må man klarlegge enkelte tilhørende attributter.

Distinguished name

SECUDE programvaren krever at CAer (og brukere) navngis vhja. et DN. Tabell 1 viser DN for CAene i UNISA når dette skrives. Organisasjonsnavnene vil være retningsgivende i valg av DN ifm. oppstart av nye CAer, men dette må undersøkes spesielt i hvert enkelt tilfelle. Enkelte organisasjoner vil kunne ha et bestemt DN fra før.

Distinguished Name
CN=EuroPKI Root Certification Authority, O=EuroPKI
OU=PCA, O=UNINETT, C=NO
O=UNINETT, C=NO
O=Norsk Regnesentral, C=NO
O=Universitetet i Tromsø, C=NO
O=Universitetet i Bergen, C=NO
O=Høgskolen i Oslo, C=NO

Tabell 1: CAer i UNISA, samt roten

E-post adresse

I X.509v3 sertifikater har man anledning til å angi e-post adresse som et alternativt navn på eieren av sertifikatet. Det anbefales sterkt å benytte dette. Man kan angi flere e-post adresser i sertifikatet. Tabell 2 viser e-post adressene for CAene i UNISA når dette skrives.

CA	E-post adresse
EuroPKI	<ingen, dessverre>
PCA	pca@uninett.no
UNINETT	uninett-ca@uninett.no
NR	nr-ca@nr.no
UIT	uit-ca@cc.uit.no
UIB	uib-ca@it.uib.no
HIOSLO	ca@adm.hioslo.no

Tabell 2: E-post adresser for CAer i UNISA, samt roten

Sertifikat URI

X.509v3 sertifikater gir også anledning til å angi en URI i sertifikater. Denne URIen skal angi hvorfra sertifikatene kan lastes ned. Tabell 3 viser de sertifikat-URIene som våre eksisterende CAer benytter. Nye CAer vil få lagt ut sine sertifikater på tilsvarende sted. Enkelte klient-programmer er i stand til automatisk å hente slike sertifikater ved behov.

CA	URI
EuroPKI	<ingen>
PCA	ldap://unisa.nr.no:1522/ou=pca,o=uninett,c=no?cacertificate
UNINETT	ldap://unisa.nr.no:1522/o=uninett,c=no?cacertificate
NR	ldap://unisa.nr.no:1522/o=nr,c=no?cacertificate
UIT	ldap://unisa.nr.no:1522/o=uit,c=no?cacertificate
UIB	ldap://unisa.nr.no:1522/o=uib,c=no?cacertificate
HIOSLO	ldap://unisa.nr.no:1522/o=hioslo,c=no?cacertificate

Tabell 3: Sertifikat URIer for CAer, samt roten

CRL Distribution Point

Lokasjoner til CRLer kan også angis i X.509v3 sertifikater. Hver CA vil jevnlig legge ut lister (CRLer) over tilbakekalte sertifikater. Dette CRL distribusjonspunktet for CAene er angitt i Tabell 4. Enkelte klientprogrammer er i stand til automatisk å hente slike CRLer

ved behov. Merk ellers at en CA ikke har sitt eget *CRL Distribution Point* i sitt eget sertifikat. Det er utsteders CRL som skal angis i sertifikatene.

CA	CRL Distribution Point
EuroPKI	http://www.europki.org/ca/root/crl/crl.der
PCA	ldap://unisa.nr.no:1522/ou=pca,o=uninett,c=no?certificaterevocationlist
UNINETT	ldap://unisa.nr.no:1522/o=uninett,c=no?certificaterevocationlist
NR	ldap://unisa.nr.no:1522/o=nr,c=no?certificaterevocationlist
UIT	ldap://unisa.nr.no:1522/o=uit,c=no?certificaterevocationlist
UIB	ldap://unisa.nr.no:1522/o=uib,c=no?certificaterevocationlist
HIOSLO	ldap://unisa.nr.no:1522/o=hioslo,c=no?certificaterevocationlist

Tabell 4: CRL URIer for CAer i UNISA, samt roten

Policy Identifier

Det er utarbeidet en policy som gir et overordnet mål på sikkerhetsnivået i EuroPKI hierarkiet. Policien angir det sikkerhetsnivået som gjelder og som underliggende CAer må forholde seg til. Policy-dokumenter identifiseres med såkalte *Policy Identifiers*, som må tilordnes spesielt. *Policy Identifier* for EuroPKI-en er vist i Tabell 5 sammen med den *policy identifier* som UNINETT PCA har hatt på sin egen policy inntil det ble bestemt at det bare skal være én policy for hele PKIen.

Merk ellers at det er sertifikatutsteders *Policy Identifier* som skal angis i et sertifikat.

CA	Policy Indentifier
EuroPKI	OID 1.3.6.1.4.1.5255.1.1.1
UNINETT PCA	OID 1.3.6.1.4.1.2428.10.1.2

Tabell 5: Policy Identifiers for UNINETT PCA og EuroPKI rot

Certificate Extensions

En CA bør initielt sørge for at de ovennevte opplysningene om CAen alltid havner i sertifikater den utsteder ved å konfigurere en spesiell *Certificate Extension*-fil. Filen heter `CA_exts` og befinner seg i CA katalogen under `secude` på CA brukerens hjemmekatalog.

Man må da sørge for at følgende er satt i filen:

```
IssuerAltName-rfc822 <CAens e-post adresse>
IssuerAltName-URI    <CAens sertifikat-URI>
DistrPoint-uRI       <CAens CRL Distribution Point>
PolicyIdentifier     <Policy OID>
```

Dersom CAen skal utstede web sertifikater (for klient eller server) for spesielle anvendelser, kan det være aktuelt også å editere på enkelte *Netscape Extensions* i `CA_exts`-filen. I tillegg må det også settes objektidentifikatorer for disse tilleggene. Dette er diskutert i detalj i avsnittene F-6 og F-7 og i Vedlegg A.

Generering av PSE på smartkort

Sørg for at det står et smartkort i leseren. CAen opprettes ved at CA-brukeren generer den vhja. `cacrt`-kommandoen i SECUDE. Dette genererer en CA katalog under `secude` på hjemmekatalogen. Ved bruk av smartkort genereres bl.a. CAens hemmelige nøkkel og et prototypesertifikat på smartkortet. Prototypesertifikatet må siden sendes som en sertifikatforespørsel til overliggende CA.

Følgene kommando genererer en CA:

```
secude> cacrt -v -c tcos:<CA-navn> -k 1024 -m <CAens e-post
        adresse> -w <CAens sertifikat-URI> "<CAens DN>"
```

Man må i prosessen angi en sikker passordfrase for beskyttelse av PSEen. *CA-navn* navngir CA katalogen som opprettes. Merk at PSEen her opprettes med default navn (`capse.cse` for CAer pr. i dag) under CA katalogen. Man trenger da ikke å spesifisere PSEen nærmere i SECUDE kommandoer.

Navnevalgene for UNISA CAer som drives av NR, fremgår av Tabell 6.

CA bruker	Hjemmekatalog	CA katalog/navn
pca	/usr/pki/pca	pca
uninett	/usr/pki/uninett	uninett
nr	/usr/pki/nr	nr
uit	/usr/pki/uit	uit
uib	/usr/pki/uib	uib
hioslo	/usr/pki/hioslo	hioslo

Tabell 6: Bruker- og katalognavn for CAer i UNISA

Generering av PSE i software

Ta evt. ut smartkort som står i leseren for sikkerhets skyld. Genereringen av PSE foregår ellers som beskrevet i forrige kapittel, med unntak av at `cacrt`-kommandoen i dette tilfellet skal være:

```
secude> cacrt -v -c <CA-navn> -k 1024 -m <CAens e-post
        adresse> -w <CAens sertifikat-URI> "<CAens DN>"
```

Eneste forskjell er altså at `tcos:` droppes i `-c` opsjonen

F-3 Sertifisering av PCAer og CAer

Generering av sertifikatforespørsler for CAer

Etter å ha generert en CA er man rede til å sende en sertifikatforespørsel til overliggende CA. I vår situasjon er det kun PCAen som må sende forespørselen eksternt. Øvrige CAer er direkte underlagt PCAen og drives alle av NR. Forespørsler og svar til og fra PCAen overføres i dette tilfellet direkte vhja. filkopiering mellom PCA katalogen og de ulike CA katalogene.

For PCAen må man notere hash-verdien for prototypesertifikatet for å bekrefte gyldigheten av sertifikatforespørselen ved behov. Verdien fremkommer når man genererer sertifikatet, og kan ellers fås fram vhja. `secude/psemaint`-kommandoene:

```
secude> psemaint -c <CA-navn>

(Angi passordfrase)

PSE> show Cert
```

Sertifikatforespørselen genereres fra SECUDE vhja. kommandoen:

```
secude> pem CERT-REQ -v -c <CA-navn> -o <cert-req-fil>
```

Cert-req-filen skal overføres til overliggende CA. PCAen sender denne forespørselen pr. e-post til EuroPKI, og må for dette diskettoverføre filen til en nettilkoplet maskin. Deretter må det avklares med EuroPKI hva slags autentisering de krever.

PCAer og CAer skal kopiere forespørseler til `archive`-katalogen som skal være opprettet i alle CA kataloger. Anbefalt filnavn for dette er:

```
archive/<CAens e-post adresse>.req
```

Utstedelse av CA sertifikat

Før et CA sertifikat kan utstedes så må det inngås en avtale om dette, jmf. Vedlegg D. Og dersom sertifikatforespørselen ikke genereres lokalt så må den verifiseres.

Etter å ha kopiert sertifikatforespørsel (*cert-req* fil) til PCA katalogen, utstedes sertifikatet av PCA-brukeren vhja. SECUDE kommandoen:

```
secude> certify -v -c tcos:pca -C TRUE -l <gyldighetsdato>
        <cert-req-fil> <output-fil>
```

Notér sertifikatets serienummer underveis.

Datoen spesifiseres på formatet *yymmddhhmmssZ*, som f eks. *010331215959Z* for 31. mars 2001 kl. 23.59.59.

Output-filen bør ha navn lik:

```
<CAens e-post adresse>.issued
```

og inneholder i tillegg til brukerens sertifikat også overliggende CAers sertifikater og tilhørende CRLer. Filen er i PEM format og kopieres til den riktige CA katalogen for installasjon og oppdatering i CAens PSE. Om CAen skal drives eksternt så sendes denne filen også tilbake til der sertifikatforespørselen kom i fra.

Sertifikater skal også legges ut tilgjengelig for distribusjons-URIer slik disse er angitt i sertifikatene. Følgende kommandoer benyttes for dette:

```
secude> psemaint -c tcos:pca (Angi passordfrase)
PSE > caissued[2] <Sertifikatets serienummer i Hex>

(Trykk underveis F for lagring til fil og angi et passende,
temporært filnavn)

PSE > q
secude> encode <filnavn> <output-fil>
```

Output-filen bør ha navnet:

```
<CAens e-post adresse>
```

Følgende to linjer skal deretter dyttes inn som hhv. første og siste linje i filen:

```
-----BEGIN CERTIFICATE-----
-----END CERTIFICATE-----
```

Hvordan sertifikatet så skal legges inn i LDAP serveren er beskrevet i avsnitt F-9.

Sertifikatforespørsler og utstedte sertifikater lagres på `archive`-katalogen til PCAen; fortrinnsvis med navn som angitt i Tabell 7.

Beskrivelse	Filnavn	Format
Sertifikatforespørsel	<CAens e-post adresse>.req	PEM
Besvart sertifikatforespørsel	<CAens e-post adresse>.issued	PEM
Distribuert CA sertifikat	<CAens e-post adresse>	CERTIFICATE

Tabell 7: Filnavn for henvendelser, svar og sertifikater som skal lagres under PCAens archive-katalog

Installasjon av CA sertifikat

PCAen mottar sitt sertifikat fra roten pr. e-post og på samme format som forespørselen; dvs. som en PEM melding i vårt tilfelle. PEM meldingen overføres fra nettilkoplet maskin til PCA maskinen og kopieres til en fil på PCA katalogen (referert som *cert-issued*-fil under). En tilsvarende fil skal tilsvarende kopieres over til CA kataloger ifm. PCA utstedelser.

PCAen/CAer installerer deretter sertifikatet i sin PSE vhja. kommandoen:

```
secude> pem -c <CA-navn> -u YES -i <cert-issued-fil>
```

PCAen/CAene må i denne prosessen verifisere at overliggende autoriteters sertifikater er ekte.

2) `causers` og `caserialumbers` kan være nyttige SECUDE kommandoer forut for `caissuedcertificate` for å finne serienummeret om det ikke ble notert underveis.

F-4 Bruk av smartkort

Det eneste å huske på ifm. med smartkort er at smartkort PSEer angis ved

```
tcos:<PSE-navn>
```

Ellers er kommandoene like.

De smartkortene vi har støtter 1024 bits nøkler og det er pr. i dag kun PCAen som ligger på smartkort.

F-5 Sertifisering av PEM brukere

Sertifisering av enkeltbrukere

Brukere sender en sertifikatforespørsel pr. e-post. Denne må diskettoverføres til UNISA maskinen, og CAen må deretter verifisere koplingen mellom identitet og forespørsel. Det er to alternativer:

- identitetsbekreftelse ved personlig fremmøte
- identitetsbekreftelse vhja. RA.

I begge tilfeller skal hash-verdien enten for den offentlige nøkkelen eller helst for prototypesertifikatet (*Certificate fingerprint*) i forespørselen presenteres for kontroll[3]. Ved bruk av RA skal RAen separat ha sendt en UNISA signert melding til CAen med navn, DN, e-post adresse og hash-verdien. Signaturen av RA-meldingen kan — når den er generert vhja. PEM — sjekkes ved:

```
secude> pem -c <CA-navn> -M 1 -i <RA-melding-fil>
```

Hash-verdien for den offentlige nøkkelen inspiseres best ved `certify`-kommando ifm. utstedelse av sertifikatet. Dersom hash-verdien ikke er korrekt kan `certify`-kommandoen avbrytes vhja. `^C` og forholdet undersøkes nærmere. Dersom man skal verifisere hash-verdien av prototype-sertifikatet[4], kan *Originator-Certificate* delen av PEM forespørselen overføres til en temporær fil (f eks. `tempfil.b64`), dekododes vhja. `decode`-kommandoen og presenteres vhja. `asn1show`:

```
% secude decode tempfil.b64 | secude asn1show
```

Et PEM brukersertifikat utstedes vhja. kommandoen:

```
secude> certify -v -c <CA-navn> -C FALSE -l
          <gyldighetsdato> <cert-req-fil> <output-fil>
```

Brukere vil normalt ikke ha spesifisert sin e-post adresse og sertifikat URI i prototypesertifikatet, slik at man underveis normalt vil måtte angi disse som *extensions* vhja:

```
Enter additional extensions: SubjectAltName-rfc822Name
                               <brukers e-post adresse>
Enter additional extensions: SubjectAltName-uRI
                               <brukers sertifikat-URI>
```

Sertifikat URI skal være på formene:

```
ldap://unisa.nr.no:1522/cn=<fornavn etternavn>,o=<CA>,c=no?
        usercertificate
ldap://unisa.nr.no:1522/cn=<fornavn etternavn>,o=<CA>,c=no?
        usersmimecertificate
```

3) *Certificate Fingerprint* legges til grunn dersom det finnes et slikt, ellers brukes *Public Key Fingerprint*. Ifm. klientsertifiseringer benyttes f.eks. ikke *Certificate Fingerprint*.

4) Merk at det ifm. `certify -v` initielt vises et prototypesertifikat til skjerm som er tilnærmet likt brukerens prototypesertifikat. Visse opplysninger, som f.eks. gyldighetsdatoene CAen spesifiserer, er imidlertid dyttet inn, og hash-verdien for `Certificate Fingerprint` vil normalt ikke samsvare med den i brukerens protorypesertifikat. Verifikasjon av `Certificate Fingerprint` gjøres altså ikke ifm. `certify` kommandoen, men som beskrevet i teksten. Verifikasjon av `Public Key Fingerprint` kan på sin side foretas ifm. `certify`.

Dersom sertifikatet skal brukes til S/MIME meldinger så benyttes den siste formen, ellers den første.

Notér sertifikatets serienummer underveis.

Output-filen bør ha navn lik:

```
<brukerens e-post adresse>.issued
```

og inneholder i tillegg til brukerens sertifikat også overliggende CAers sertifikater og tilhørende CRLer. Filen er i PEM format og skal diskettoverføres til nettilkoplet PC og returneres pr. e-post til brukeren for installasjon og oppdatering i brukerens PSE.

Sertifikat skal gjøres tilgjengelig for søke-tjenestene i UNISA og for distribusjons-URIene i sertifikatene. Dersom man har notert serienummeret[5] får man fram sertifikatet slik:

```
secude> psemaint -c <CA-navn> (Angi passordfrase)
PSE > caissued <Sertifikatets serienummer i Hex>

(Trykk underveis F for lagring til fil og angi et passende,
temporært filnavn)

PSE > q
secude> encode <filnavn> <output-fil>
```

Output-filen bør ha navnet:

```
<brukerens e-post adresse>
```

Følgende to linjer skal deretter legges inn som hhv. første og siste linje i filen:

```
-----BEGIN CERTIFICATE-----
-----END CERTIFICATE-----
```

Hvordan sertifikatet til slutt legges inn i LDAP serveren er beskrevet i F-9.

Sertifikatforespørsler, RA-verifikasjoner og utstedte sertifikater lagres på `archive` katalogen til CAen; fortrinnsvis med navn som angitt i Tabell 8.

Beskrivelse	Filnavn	Format
Sertifikatforespørsel	<brukers e-post adresse>.req	PEM
RA-verifikasjon	<brukers e-post adresse>.ra	PEM
Besvart sertifikatforespørsel	<brukers e-post adresse>.issued	PEM
Distribuert brukersertifikat	<brukers e-post adresse>	CERTIFICATE

Tabell 8: Filnavn for henvendelser, svar og sertifikater som skal lagres under CAens archive-katalog ifm. PEM brukersertifisering

5) Har man ikke serienummeret for hånden så kan SECUDE kommandoene `causers` og `caserialnumbers` være nyttige.

F-6 Sertifisering av web og S/MIME klienter

Det følgende dekker sertifisering av Netscape og Explorer klienter. Disse sertifikatene kan benyttes både for SSL og S/MIME. Når det gjelder Explorer så er det Explorer 3.* som er testet. Det var dette vi testet innledningsvis og planen var å teste nyere versjoner etterhvert som behov oppstod. Det har imidlertid ikke oppstått noe behov.

Sertifisering av Netscape klienter

Forespørselen mottas pr. e-post og diskettoverføres til CA maskinen. Forespørselen består av utfylt informasjon pluss en base64-kodet blokk med "`-----BEGIN NEW CERTIFICATE REQUEST-----`" (hode) og "`-----END NEW CERTIFICATE REQUEST-----`" (hale) som illustrert i Figur 4. Hele forespørselen arkiveres ihht. Tabell 8. Blokken minus hode og hale kopieres til en fil og skal benyttes som input til sertifiseringskommandoen.

```
Fullt navn person:      Shahrzade Mazaher
E-post:                 mazaher@nr.no
Telefon:                22852593

Fullt navn organisasjon:Norsk Regnesentral
Kortnavn organisasjon:  NR
Land:                   NORGE

Organisasjonsenhet:

Klienttype:             Mozilla/4.61 [en] (X11; I; SunOS 5.7 sun4u)

-----BEGIN NEW CERTIFICATE REQUEST-----
MIG6MGYwXDANBgkqhkiG9w0BAQEFAANLADBIAkEA05jRickN5rdbYm80QHgBaJF0
YNnm7jl/BjSg5VW3FxD2XeBZWLXhyvCrHyu4M10CMxfbgX+wi2OWe0uPRm5gTQID
AQABFgZwYXNzd2QwDQYJKoZIhvcNAQEEBQADQQAyEDypsQxMnJ1xxsDheGccoM8H
Li2OVYa5ml7G0Pva8mW77mEvD1tPEExKprjqr8zZ0GavC5QqFGuqwyeTEm1T
-----END NEW CERTIFICATE REQUEST-----
```

Figur 4: Eksempel på mottatt sertifiseringsforespørsel for web-klient der UNISAs web-sider er benyttet for generering av forespørselen.

Koplingen mellom identitet og forespørsel må deretter verifiseres, analogt med det som er beskrevet i avsnitt F-5. Merk at klientforespørsler inneholder fingerprint av den offentlige nøkkelen, men ingen *Certificate Fingerprint* som en PEM basert forespørsel. Fingerprinten for den offentlige nøkkelen kan tilsvarende benyttes som grunnlag for verifikasjon. Normalt vil imidlertid brukere initiere nøkkelgenereringen og sertifikatforespørselen fra UNISAs web-sider. I dette tilfellet presenteres en hash-verdi av forespørselen tilbake til brukeren, som han siden skal presentere for sin RA eller CA. Utsteder beregner på sin side en hash av forespørselen (minus hode og hale) vhja. skriptet `hashcheck` (se F-12) og sammenlikner med den oppgitt for brukeren.

Forespørselen sertfiseres vhja. kommandoen:

```
secude> certify -v -c <CA-navn> -C FALSE -l
        <gyldighetsdato> -e TRUE -t CERTIFICATE
        -B "-----BEGIN CERTIFICATE-----"
        -B "-----END CERTIFICATE-----"
        <input-fil> <ouput-fil>
```

Kontroller bl.a. at DN blir riktig satt. Et DN skal angis i rekkefølgen:

```
CN= , [OU= ,] O=, , C=NO
```

Eventuelle e-post adresser angis eksplisitt vhja:

```
Enter additional extensions: SubjectAltName-rfc822Name
                              <brukers e-post adresse>
```

Og sertifikatets URI må angis slik:

```
Enter additional extensions: SubjectAltName-uRI
                              <brukers sertifikat-URI>
```

Distribusjon av sertifikatet gjøres som beskrevet i F-5 og F-9. Og brukeren vil hente sertifikatet ved å benytte arkivtjenesten på UNISAs web-sider.

Dersom det er behov for å avgrense anvendelsen av sertifikatet til kun spesielle anvendelser, f.eks. bare for S/MIME eller bare for SSL, så kan man benytte `-E` opsjonen i ovennevnte `certify` kommando og angi en bestemt *extension block*. `CA_exts`-filer under CA kataloger inneholder de nødvendige *extension blocks*, og de ulike Netscape *extensions* angis under en av disse blokkene med ønskede verdier.

NB: Merk at dersom sertifikatet skal benyttes for kodesignering, så må dette angis eksplisitt vhja. Netscape-*extensions*, siden default i Netscape-applikasjoner er ikke å tillate kodesignering.

Med SECUDE kan to Netscape *extensions* settes pr. i dag:

- netscape-cert-type
- netscape-comment

Disse har tilhørende objektidentifikatorer som må settes i en fil `objids` under `$SECUDE_ETC` som følger:

```
netscape-cert-type    2 16 840 1 113730 1 1
netscape-comment      2 16 840 1 113730 1 13
```

Netscape-comment er bare en streng som vises ved presentasjon av sertifikatet i Netscape. *Netscape-cert-type* er en 8-bits kode som angir de nevnte anvendelsene av sertifikatet i ihht. Tabell 9.

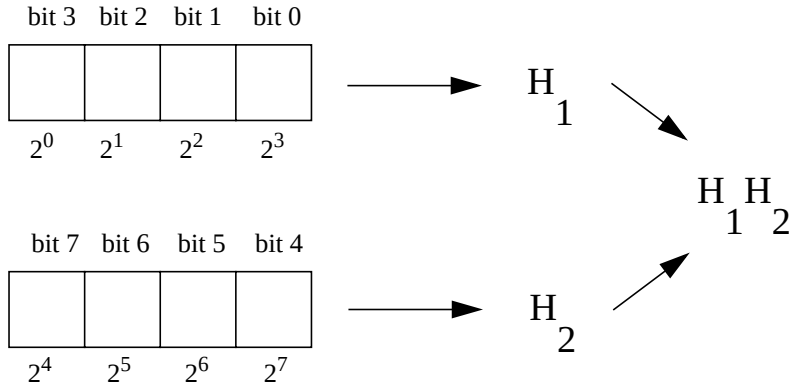
Bit nr.	Betydning
0	SSL klient
1	SSL server
2	S/MIME klient
3	Kodesignering
4	Reservert for framtidig bruk
5	SSL CA
6	S/MIME CA

Tabell 9: Netscape cert types

Bit nr.	Betydning
7	Kodesignering CA

Tabell 9: Netscape cert types

Bitverdi ’1’ indikerer ja’ og verdi ’0’ indikerer ’nei’. I SECUDE angis den samlede koden ved et tosifret heksadesimalt tall, der venstre heksadesimale tall angir bitverdiene på bit0-bit3 speilet, mens høyre heksadesimale tall angir bitverdien på bit4-bit7 speilet. Dette er illustrert i Figur 5:



Figur 5: Angivelse av heksadesimal bitkode for netscap-cert-type i Secude

For å angi et klientsertifikat som f.eks. kan (og kun!) benyttes for SSL, S/MIME og kodesignering, må man skru på bit 0, 2 og 3 ved å angi koden ’`B0`’ (`B` = $2^3+2^1+2^0$) for *netscape-cert-type* under *NetscapeClient*-blokken i `CA_exts`-filen og så benytte opsjonen ’`-E NetscapeClient`’ i `certify`. Følgende kommando kan da benyttes:

```
secude> certify -v -c <CA> -C FALSE -l <utløpsdato>
        -e TRUE -t CERTIFICATE -E NetscapeClient
        -B "-----BEGIN CERTIFICATE -----"
        -B "-----END CERTIFICATE-----"
        <innfil> <utfil>
```

Henvendelser og sertifikater bør lagres på CAens `archive` katalog med navn som angitt i Tabell 10.

Beskrivelse	Filnavn	Format
Sertifikatforespørsel	<navn>-cl@<domene>.req	CERTIFICATE
RA verifikasjon	<navn>-cl@<domene>.ra	PEM
Besvart sertifikatforespørsel	<navn>-cl@<domene>.issued	CERTIFICATE
Distribuert klientsertifikat	<navn>-cl@<domene>	CERTIFICATE

Tabell 10: Filnavn for henvendelser og sertifikater som skal lagres under CAens archive-katalog ifm. Netscape klientsertifisering

Sertifisering av Explorer 3.* klienter

Sertifisering av Microsoft Internet Explorer klienter foregår analogt med det som er beskrevet i forrige kapittel, men dessverre ikke helt likt. Videre har Microsoft implementert sertifikatstøtten helt forskjellig i versjon 3.* og versjon 4.* av Explorer. Beskrivelsen under vil fungere med versjon 3.*.

Som svar på en sertifikatforespørsel, forlanger Explorer sertifikatet returnert i base64 kodet PKCS 7 format[6]. Explorer støtter ikke CRLer, og svaret fra utsteder med klientsertifikatet kan ikke inneholde CRLer, hvilket det vil gjøre ved bruk av `certify` kommandoen. Utsteder må derfor i sin PSE midlertidig omdøpe sine `crls.dir` og `crls.pag` filer før utstedelse av Explorer sertifikater. Svaret må videre inneholde den *SessionID* som blir generert på klientsiden ifm. sertifikatforespørsler i Explorer.

Forespørselen mottas pr. e-post og diskettoverføres til CA maskinen. Forespørselen består av utfylt informasjon pluss en base64-kodet blokk med "`-----BEGIN CERTIFICATE-----`" og "`-----END CERTIFICATE-----`" som hode og hale. Hele forespørselen arkiveres ihht. Tabell 10. Blokken minus hode og hale kopieres til fil og skal benyttes som input til sertifiseringskommandoen. Forespørselen skal også inneholde den nevnte *SessionID*-en. Koplingen mellom identitet og forespørsel må verifiseres som tidligere forklart.

Forespørselen inspiseres og sertfiseres vhja. kommandoen:

```
secude> certify -v -c <CA-navn> -C FALSE -l
        <gyldighetsdato> -e TRUE
        -B "-----BEGIN CERIFICATE-----"
        -B "-----END CERIFICATE-----"
        <input-fil> <ouput-fil>
```

Kontroller bl.a. at DN blir riktig satt. Et DN skal angis i rekkefølgen:

```
CN= , [OU= ,] O=, , C=NO
```

Eventuelle e-post adresser angis eksplisitt vhja:

```
Enter additional extensions: SubjectAltName-rfc822Name
                              <brukers e-post adresse>
```

Og sertifikatets URI må angis slik:

```
Enter additional extensions: SubjectAltName-uRI
                              <brukers sertifikat-URI>
```

Distribusjon av sertifikatet gjøres som beskrevet i F-5 og F-9. Og brukeren vil hente sertifikatet ved å benytte arkivtjenesten på web-sidene.

Dersom det er behov for å avgrense anvendelsen av sertifikatet til kun spesielle anvendelser, kan man benytte `-E` opsjonen som beskrevet under sertifisering av Netscape klienter. Explorer skal kunne støtte Netscape *extensions*.

6) Netscape håndterer de fleste formater. Vi returnere typisk sertifikater på base64 kodet ASN.1 format

Henvendelser og sertifikater bør lagres på CAens `archive` katalog med navn som angitt i Tabell 11.

Beskrivelse	Filnavn	Format
Sertifikatforespørsel	<navn>-ie@<domene>.req	CERTIFICATE
RA verifikasjon	<navn>-ie@<domene>.ra	PEM
Besvart sertifikatforespørsel	<navn>-ie@<domene>.issued	CERTIFICATE
Distribuert klientsertifikat	<navn>-ie@<domene>	CERTIFICATE

Tabell 11: Filnavn for henvendelser og sertifikater som skal lagres under CAens archive-katalog ifm. Explorer klientsertifisering

F-7 Sertifisering av web servere

Forespørselen mottas pr. e-post og diskettoverføres til CA maskinen. Forespørselen består av en base64-kodet melding med "`-----BEGIN/END NEW CERTIFICATE REQUEST-----`" som hode og hale. Disse to linjene fjernes. Koplingen mellom server og forespørsel må deretter verifiseres, analogt med det som er beskrevet tidligere. Merk at serverforespørsler ikke inneholder noe *Certificate Fingerprint* og verifikasjonen må derfor basere seg på fingerprint av den offentlige nøkkelen.

```
Fullt navn person:       Jonn Skretting
E-post:                  skr@nr.no
Telefon:                 22852625

Fullt navn organisasjon: Norsk Regnesentral
Kortnavn organisasjon:   NR
Land:                    NORGE

Organisasjonsenhet:      OMNI

Servertype:              APACHE
Servernavn:              www.omni.nr.no

-----BEGIN NEW CERTIFICATE REQUEST-----
MIICJzCCAZACAQAwDQYJKoZIhvcNAQEEBQAwXDELMAkGA1UEBhMCTk8xEzARBgNV
BAgTClNvbWUtU3RhdGUxCzAJBgNVBAoTAk5SMREwDwYDVQQDEwhKb25uIFNrcjEY
MBYGCSqGSIb3DQEJARYJc2tyQG5yLm5vMB4XDTk5MDkyNzA5MTUwNloXDTk5MTAy
NzA5MTUwNlowXDELMAkGA1UEBhMCTk8xEzARBgNVBAgTClNvbWUtU3RhdGUxCzAJ
BgNVBAoTAk5SMREwDwYDVQQDEwhKb25uIFNrcjEYMBYGCSqGSIb3DQEJARYJc2ty
QG5yLm5vMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDfrtFVvWs8KCkVMgUw
SdBcg6CzfTu2E1G5DdDZjOug7Jy6hcoCsacxQ3t1QH7CpvdV8wLlO05Wd2LhcNbz
+qSipFAkfxKyEhgC9pPMRPDQj1tgeY8Ms8BQTJlpY1L8iJTp3TovLXElG7nxAZRE
+g8miv6LKmOeN1Bc67B4N2Ao1QIDAQABMA0GCSqGSIb3DQEBBAUAA4GBAFP1X0CF
lU32Na+y2Mn2elu2jXxOgkUf5xPSX+Gl3Hrd/DJqLzd951lHStHDs9eIIvL0j0rX
uRMksq9c5B2gtn2Z+M4DvHP3uz0xmXLRAAqJCCYGUHSZKv+OodXe7HjzKwHoQj8e
nC/poKcDkfzo6z/APsVMZmROVRVj+vcc2jSy

-----END NEW CERTIFICATE REQUEST-----
```

Figur 6: Eksempel på mottatt sertifiseringsforespørsel for web-server der UNISAs web-sider er benyttet for generering av forespørselen.

Forespørselen inspiseres og sertifiseres vhja. kommandoen:

```
secude> certify -v -c <CA-navn> -C FALSE -l
        <gyldighetsdato> -e TRUE -t CERTIFICATE
        -B "-----BEGIN CERIFICATE-----"
        -B "-----END CERIFICATE-----"
        <input-fil> <ouput-fil>
```

Output-filens anbefalte filnavn er angitt under.

Et DN må etableres for serveren. E-post adresse og sertifikat URI må angis eksplisitt vhja:

```
Enter additional extensions: SubjectAltName-rfc822Name
                              <brukers e-post adresse>
Enter additional extensions: SubjectAltName-uRI
                              <brukers sertifikat-URI>
```

Dersom sertifikatet ikke skal distribueres via LDAP så behøver ikke `uRI` ekstensjonen å være med[7]. Ellers så bør den være på formen som er oppgitt i F-5, der `cn=<fornavn etternavn>` erstattes med `cn=<servernavn>`. Et eksempel på et servernavn er *www.nr.no*.

Man kan eventuelt inspisere forespørselen separat i forkant ved kommandoen:

```
% secude decode input-fil | secude asn1show
```

Det utstedte sertifikatet diskettoverføres til nettilkoplet maskin og legges tilgjengelig for UNISAs web søketjeneste, jmf. F-9, dersom det skal distribueres.

Dersom det er behov for å avgrense anvendelsen av sertifikatet til kun spesielle anvendelser, kan man benytte `-E` opsjonen i ovennevnte `certify` kommando som antydet tidligere.

Henvendelser og sertifikater bør lagres på CAens `archive` katalog med navn som angitt i Tabell 12.

Beskrivelse	Filnavn	Format
Sertifikatforespørsel	<navn>-server@<domene>.req	CERTIFICATE
RA verifikasjon	<navn>-server@<domene>.ra	PEM
Besvart sertifikatforespørsel	<navn>-server@<domene>.issued	CERTIFICATE
Distribuert serversertifikat	<navn>-server@<domene>	CERTIFICATE

Tabell 12: Filnavn for henvendelser og sertifikater som skal lagres under CAens archive-katalog ifm. serversertifisering

7) Men policien sier at alle sertifikater skal distribueres med mindre eieren eksplisitt motsetter seg det.

F-8 Administrasjon av tilbakekallingslister

Tilbakekalling av sertifikater

Sertifikater kan tilbakekalles som følge av at en bruker slutter eller at de tilhørende private nøklene mistenkes å være kompromittert. I begge tilfeller skal den utstedende CAen varsles vhja. en signert melding. Alternativt kan CAen varsles og kontrollere hvorvidt opplysningen er korrekt på annen sikker måte.

Signaturer på PEM meldinger verifiseres vhja:

```
secude> pem -c <CA-navn> -M 1 -i <signert-melding-fil>
```

Ifm. kompromitteringer, eller mulige sådanne, vil man måtte undersøke omstendighetene spesielt.

Et sertifikat tilbakekalles vhja. kommandoen:

```
secude> psemaint -c <CA-navn>

(Angi passordfrase)

PSE> revoke <Serienummer>
```

Det tilhørende sertifikats serienummer kan finnes bl a. vhja.:

```
PSE> caserialnumbers[8] <name>
```

Dette sertifikatet vil dermed bli angitt i neste CRL som CAen utsteder og distribuerer.

Utstedelse av tilbakekallingslister

Hver CA skal månedlig (minimum hver 40 dager ihht. EuroPKI policy) utstede og distribuere nye CRLer. Dette gjøres som følger:

```
secude> psemaint -c <CA-navn> (Angi passordfrase)
PSE> caprolong

(Tast inn dato 'relativt' og angi 0 years, 0 months og 30 eller
31 days. Timer, minutter etc trengs ikke å spesifiseres. Svar
'yes' for å signere CRLen til slutt)

PSE> q
secude> pem CRL -c <CA-navn> -o <crl-fil>
```

der *crl-filen* f.eks. kan ha navn `crl@<domene>`. Vi er imidlertid ikke interesserte i hele PEM meldingen, CRL-en er kun den biten som kommer etter *CRL:* i PEM formatet, jmf. Figur 7. Disse linjene trekkes derfor ut og det er de som representerer den base64 kodede input-en til distribusjonstjenesten omtalt i neste avsnitt.

Ettersom vi utsteder fulle CRLer, og ikke bare inkrementelle, så er det kun nødvendig å lagre den siste utstedte CRLen for en CA, og denne kan overføres til `archive` katalogen.

8) `causers` kan være en nyttig SECUDE kommando forut for `caserialnumbers`.

```
-----BEGIN PRIVACY-ENHANCED MESSAGE-----
Proc-Type: 4,CRL
Content-Domain: RFC822
CRL:
 MIIBcjCB3AIBATANBgkqhkiG9w0BAQQFADAtMQswCQYDVQQGEwJOTzEQMA4GA1UE
 ChMHVU5JTkVUVDEMMAoGA1UECxMDUENBFw0wMDEwMjMxNTQ2MTRaFw0wMDExMjUx
 NTQ2MTRaoHsweTAfBgNVHSMEGDAWgBQAvJ6iU4Ic2xCnqM1mJEU+YzVTKDBWBgNV
 HRIETzBNgQ5wY2FAdW5pbmV0dC5ub4Y7bGRhcDovL3VuaXNhLm5yLm5vOjE1MjIv
 b3U9cGNhLG89dW5pbmV0dCxjPW5vP2NhY2VydGlmaWNhdGUwDQYJKoZIhvcNAQEE
 BQADgYEAKt4St4v2ZMizrB1xp8WKfwhdX/jwdCsb5NeljNSYPWEZ8qJxU+FShXGN
 +GYuBr8ApoIdEe58/4bibRZnskdSbNK2r7TEMszpSOkvKkmGj6i9Nko2TjoB1s20
 PJbH+ZZ8S9ozqNZcHz/INj5a/agmXO00SM+xuXhlGH9VS18UVcI=
Originator-Certificate:
 MIIDsTCCApmgAwIBAgIBAjANBgkqhkiG9w0BAQQFADBBMRAwDgYDVQQKEwdFdXJv
 UEtJMS0wKwYDVQQDEyRFdXJvUEtJIFJvb3QgQ2VydGlmaWNhdGlvbiBBdXRob3Jp
 dHkwHhcNMDAwMTA1MTQyMjEzWhcNMDEwMzMxMjE1OTU5WjAtMQswCQYDVQQGEwJO
 TzEQMA4GA1UEChMHVU5JTkVUVDEMMAoGA1UECxMDUENBMIGfMA0GCSqGSIb3DQEB
 AQUAA4GNADCBiQKBgQDtblEMzswRuqN6KjwQxifqLSI5C7aoyLDVfgHYngH0E1Ka
 Q/veDo8kl77FVsrOrHWregasEvMdcVlI3oaTaZVnsm5Iv1LPPnc5TlBdbcaZqFmG
 Z6XfqCsECfw7epF6be8VekV2YGwLFvGjw/L+v20s7PJUG/6Wxb7kNC2zZnW/wQID
 AQABo4IBSjCCAUYwHwYDVR0jBBgwFoAUjNyLsaVKkOdOiHMYPJ3VXn7kss0wHQYD
 VR0OBBYEFAC8nqJTghzbEKeozWYkRT5jNVMoMA4GA1UdDwEB/wQEAwIB9jBOBgNV
 HSAERzBFMEMGCisGAQQBqQcBAQEwNTAzBggrBgEFBQcCARYnaHR0cDovL3d3dy5l
 dXJvcGtpLm9yZy9jYS9yb290L2Nwcy8xLjEvMFYGA1UdEQRPME2BDnBjYUB1bmlu
 ZXR0Lm5vhjtsZGFwOi8vdW5pc2EubnIubm86MTUyMi9vdT1wY2Esbz11bmluZXR0
 LGM9bm8/Y2FjZXJ0aWZpY2F0ZTAPBgNVHRMBAf8EBTADAQH/MDsGA1UdHwQ0MDIw
 MKAuoCyGKmh0dHA6Ly93d3cuZXVyb3BraS5vcmcvY2Evcm9vdC9jcmwvY3JsLmRl
 cjANBgkqhkiG9w0BAQQFAAOCAQEAGtXSQTfMcDF7D8UHlI9KAS0FGzV7j+aFdZEE
 3a/4qcfSPciZEUIhIxutyaiZ7ccfFmP0ZBzjDZ4vYA3bXG0w4GS6S3f3v6Gy31tH
 I5z3EOKYdM9TCYiSPGJHOltBTAq4+q7NXiVK59Wm2pLENsrkAf4RNYl9KSUjXvxb
 errmMw+qmwPOx6cLkbPyqLaimFNVZ28K3pTSuMq2Iglg1LHdplZ5fRzWtPK8NZpG
 tODvDjoViC6Ed205WUdMYwPkt8pOjHk2e/+QY+rA5kaWT9ssc0Lej3VB+M2ZN72O
 KgQ3uTE/dr3qH+PRsEIH5HTxjkSLC+tPgPyZH2lZXnj4H9ggdQ==
-----END PRIVACY-ENHANCED MESSAGE-----
```

Figur 7: CRL-biten av en PEM header markert med fet type.

CRL filene diskettoverføres til nettilkoplede maskiner og legges tilgjengelig for søketjenesten; ihht. hva som er beskrevet i F-9. CRLen til roten er tilgjengelig i flere formater fra:

`http://www.europki.org/ca/root/crl/en_index.html`

F-9 Distribusjon av sertifikater og tilbakekallingslister

Ved innlegging av sertifikater og tilbakekallingslister i LDAP serveren så skal disse alltid være på ASN.1 format. Om sertifikater eller tilbakekallingslister kommer fra CAen på kodet base64 form, så må de dekodes. Dette gjøres ved:

```
% secude decode <fil.b64> <fil.asn>
```

Om de to linjene:

```
-----BEGIN CERTIFICATE-----
-----END CERTIFICATE-----
```

er med i `fil.b64`, så må disse fjernes før dekodingen.

For å sjekke at innholdet på `fil.asn` er bra, så kan man bruke SECUDE til å vise fram sertifikatet:

```
% secude asn1show <fil.asn>
```

Skrives ikke innholdet pent og pyntelig ut, så må en ny versjon av innholdet skaffes til veie. (Ta utgangspunkt i resultatet fra `pem CRL` kommandoen omtalt i avsnitt F-8. Det er ikke nødvendig å utstede ny CRL som sådan.)

bin/tcldish

Denne LDAP klienten benyttes for å manipulere på innholdet i LDAP serveren. Vanlige og nyttige kommandoer er:

- `dbulkdump c=no -file <filnavn>`
- `dbulkload -ldif file.ldif`
- `dmoveto o=someOrg,c=no`
- `dshow`
- `dshow “<fullt DN>”`
- `dshow <nummer som vist ved dlist>`
- `dlist`
- `dshowname`
- `dsearch -subtree “(o=Univ*)”`
- `ddelete (sletter hele entrien)`
- `dadd “<cn=MyName,o=MyOrg,c=no>”`
  `leser ~/.dishdraft`
- `dmodify -add “cACertificate={FILE}/.../cert.asn”`
- `dmodify -add “certificateRevocationList={FILE}/.../crl.asn”`
- `dmodify -remove userCertificate`

Eksempler på bruk finnes nedenfor. Ved innlegging av nye entries er det `dadd` som som leser `~/.dishdraft` som benyttes.

Innlegging av ny organisasjon

`~/.dishdraft` skal se noenlunde slik ut:

```
objectClass= top
objectClass= organization
objectClass= certificationAuthority
o= nr
o= Norsk Regnesentral
mail= krypto@nr.no
cACertificate={FILE}/local/unisa/…/ca.asn
certificateRevocationList={FILE}/local/unisa/…/hiof/crl.asn
```

Den ene `o`-en er den som tilsvarer organisasjonens mailadresse, den andre er det fulle navnet slik som i DN-et i sertifikatet. Dersom de er like (som for *Uninett*) så tas bare ett organisasjonsnavn med. `Mail` er mailadressen til CAen (altså ikke til organisasjonen).

Innlegging av ny organisasjonsenhet

PCAen er en OU til o=Uninett så vi bruker den som eksempel. `~/.dishdraft` skal da se slik ut:

```
objectClass= top
objectClass= organizationalUnit
objectClass= certificationAuthority
ou= PCA
mail= pca@uninett.no
cACertificate= {FILE}/local/unisa/…/cert.asn
certificateRevocationList= {FILE}/local/unisa/…/crl.asn
```

Den er altså ikke vesens forskjellig fra slik den ser ut ved innlegging av ny organisasjon.

Innlegging av ny person

`~/.dishdraft` skal se noenlunde slik ut:

```
objectClass= top
objectClass= person
objectClass= organizationalPerson
objectClass= inetOrgPerson
cn= Fornavn Etternavn
surname= Etternavn
mail= bruker@org.no
description= SessionID 783111572265625
userCertificate= {FILE}/local/unisa/…/cert.asn
userSMIMECertificate= {FILE}/local/unisa/…/smimecert.asn
```

description fylles ut i tilfellet av MSIE klienter, ellers tas den bort. For den aktuelle tallverdien, jmf. F-6 og *"Sertifisering av Explorer 3.* klienter"*.

userSMIMECertificate benyttes for sertifikater som skal benyttes til S/MIME meldinger, *userCertificate* benyttes for andre sertifikater.

Innlegging av ny web-server

Web-server sertifikater har ikke vært distribuert i UNISA. Policien som sier at absolutt alle sertifikater må distribueres med mindre at eieren eksplisitt motsetter seg det, ble først vedtatt helt på slutten av ICE-CAR. Motivasjonen bak ikke å distribuere web-server sertifikater vhja. LDAP har vært at sertifikatene benyttes i SSL sammenheng og SSL proto-

kollen spesifiserer at en web-server alltid skal svare med sitt sertifikat når den kontaktes av en klient. Derfor skulle det ikke være nødvendig for noen å hente et web-server sertifikat fra en egen distribusjonstjeneste. Når det så allikevel er tatt inn i policien, så er det av generelle hensyn og for å gjøre det mulig for de som vil se på sertifikatet å hente dette fra det stedet hvor alle andre sertifikater finnes.

Når man skal lage et kataloginnslag for å distribuere web-serversertifikater så bør man velge objektklasse slik at man får med rimelige attributter. EuroPKIs italienske CA har definert sin egen objektklasse `poliServer` med attributtene `cn` (navnet på serveren, eksempelvis *www.europki.org*), `mail` for mailadressen til web-master, `organization` for vertsorganisasjonen, `departement` for avdeling, `host` for vertsmaskin og `userCertificate` til å holde selve sertifikatet.

Oppdatering av attributter

Dette gjøres ved å benytte varianter av `tcldish` kommandoen `dmodify`.

Om vi skulle ønske å legge inn ny CRL for PCAen så kunne vi f.eks. gjøre følgende:

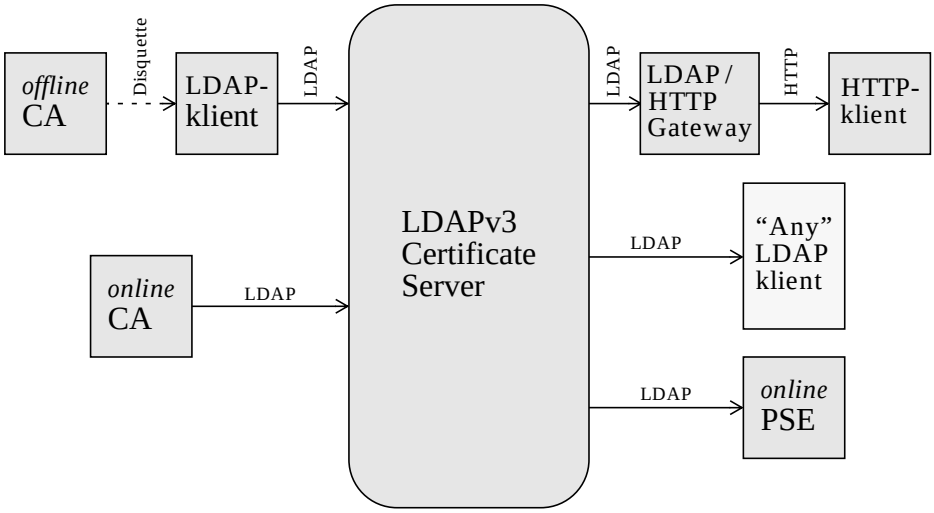
```
unisa-skr> bin/tcldish
Welcome to TclDish !
Enter password for "<cn=DSA Manager, cn=X509v3 Directory,
o=Uninett, c=NO>": <Passord>

TclDish% dmove c=no
TclDish% dlist
1     o=Uninett
2     o=hioslo
3     o=uit
4     o=uib
5     o=nr
6     o=hiof
TclDish% dmove 1
TclDish% dlist
7     cn=X509v3 Directory
8     ou=PCA
9     cn=Anders Lund
10    cn=Bente Myrset
11    cn=Hilde Hopen
12    cn=Ingrid Melve
13    cn=Trond Skjesol
14    cn={T.61}Magnus Str\\F8mdal
TclDish% dmove 8
TclDish% dshow
… (output not shown here, but everything is there)
TclDish% dmodify -remove certificateRevocationList
TclDish% dshow
… (output not shown here, certificateRevocationList is gone)
Modified <ou=PCA, o=Uninett, c=NO>
TclDish% dmodify -add "certificateRevocationList=\
          {FILE}/local/home/skr/Unisa/tmp/pca.crl"
Modified <ou=PCA, o=Uninett, c=NO>
TclDish% dshow
```

```
… (output not shown here, new certificateRevocationList is in)
TclDish% dmove .. (vil manøvrere en entry opp i DITet)
TclDish% dquit
unisa-skr>
```

Sertifikatdistribusjon med LDAP

Sertifikatdistribusjonen er lagt opp med det for øye at den skal fungere som vist i figuren under. Når sertifikater hentes ut vhja. UNISAs web sider så skjer dette vhja. CGI scripts som benytter PerlDAP. Vi har ingen on-line CAer i UNISA.



Figur 8: Sertifikatdistribusjon med LDAP

F-10 Autorisering av RAer

Se policy og practice statements utarbeidet for tjenesten og gjengitt i Vedlegg B og Vedlegg C.

F-11 Backup-rutiner

Små backup-mengder kan tas til diskett. Større mengder tas på CD eller tape.

F-12 Scriptet hashcheck for fingerprintverifisering

```
#!/local/bin/perl

if ($#ARGV != 0 || ($#ARGV == 0 && $ARGV[0] eq '-h')) {
  print "\nUsage: hashcheck \<inputFile>\n";
  print "\<inputFile\> skal være uten start/slutt
           direktiver.\n\n";
  exit -1;
}

$request = $ARGV[0];
$secude = "/local/packages/<secude-versjon>/bin/secude";

$cmd = "cat $request | $secude decode | $secude hsh -a sha1 |
          $secude encode -x";

open (FPRNT, "$cmd |") || die "Can't open $request";
$newCert = <FPRNT>;
close (FPRNT);

$newCert =~ s/[0-9,A-F]{2}/$&:/g;
chop $newCert;
chop $newCert;

print "\nHashverdi: $newCert\n\n";
```

F-13 Sertifisering av PGP brukere

I PGP-tjenesten ble bare CAer sertifisert; dvs. hver enkelt PGP-bruker var å oppfatte som en CA. Det følgende dokumenterer hvordan UNINETT CA signerte/sertifiserte nøkler for PGP brukere. Se for øvrig policy og practice statements utarbeidet for PGP tjenesten og gjengitt i Vedlegg E.

1. Sertifiseringsprosessen initieres ved at en PGP-bruker (*pgp-bruker*) sender UNINETTs PGP-administrator (*pgp-ca*) to fullstendig utfylte og undertegnede avtaleskjemaer pr. post. Prosessen avbrytes dersom enkelte av de etterfølgende kontroller eller punkter ikke oppfylles.
2. *pgp-ca* undertegner avtaleskjemaene og returnerer det ene pr. post til *pgp-bruker*.
3. Via e-post[9] sender *pgp-bruker* sin offentlige PGP-nøkkel, sitt fulle navn og e-postadresse, (evt. også organisasjon) i et PGP-sertifikat til *pgp-ca*. Navn og e-postadresse identifiserer *pgp-bruker* og spesifiseres i sertifikatet som:
   `Navn [, Organisasjon] <epostadresse>`
4. *pgp-ca* kontrollerer off line *pgp-bruker* sin identitet og koplingen til oversendt nøkkel. Dette bør gjøres ved at *pgp-ca* kontakter *pgp-bruker* telefonisk og ber *pgp-bruker* fakse over navn, kopi av godkjent ID-kort[10] og fingerprint av oversendt

9) til *pgp-ca@uninett.no*

nøkkel.

5. *pgp-ca* diskettoverfører nøkkelen i punkt 3. til CA-maskinen hvor den lagres som `<navn>.asc` på PGP-området[11]. `<navn>` er *pgp-bruker* sitt navn. *pgp-ca* må kopiere PGP-sertifikatet, dvs. "BEGIN PGP PUBLIC KEY BLOCK *innhold* END PGP PUBLIC KEY BLOCK"-blokken, over til diskett, logge seg på CA-maskinen som *pgp* og kopiere filen på plass vhja:
   ```
   pgp% cd /usr/home/pgp/cas
   pgp% mcopy a:<diskettfil> <navn>.asc
   ```
6. *pgp-ca* kan nå signere *pgp-bruker* sine PGP-nøkler vhja. kommandoen:
   ```
   pgp% pgp -ka <navn>.asc
   ```
   Normalt skal man svare *ja* på de spørsmålene som kommer opp. Husk spesielt å sammenligne fingerprintverdien med den i punkt 4. Kommandoen resulterer i at *pgp-bruker* sin nøkkel dyttes signert inn i *pgp-ca* sin nøkkelring.
7. *pgp-ca* legger den signerte nøkkelen ut på UNINETTs web-server. Dette gjøres ved at *pgp-ca* henter nøkkelen fra nøkkelringen vhja:
   ```
   pgp% pgp -kx <navn/ID> <navn>.pgp
   ```
   Utfilen `<navn>.pgp` diskettoverføres til ftp-området[12] og er dermed tilgjengelig for web-serveren. I kommandoen angir `<navn/ID>` navn-/subject-feltet i PGP-sertifikatet og `<navn>` samme navn som angitt i `<navn>.asc`.
8. *pgp-ca* sender den signerte PGP-nøkkel via e-post til *pgp-bruker*. Dette gjøres tilsvarende ved at *pgp-ca* henter nøkkelen ut av nøkkelringen i ascii-form:
   ```
   pgp% pgp -kxa <navn/ID> <navn>.asc
   ```
   diskettoverfører utfilen `<navn>.asc` og sender denne via e-post.
9. *pgp-ca* ber *pgp-bruker* signere PGP-nøkkelen som er utlagt på web, og sende denne signert tilbake med e-post.
10. *pgp-ca* diskettoverfører denne siste mottatte nøkkelen til CA-maskinen på PGP-området, verifiserer signaturen og dytter den ned i nøkkelringen vhja:
    ```
    pgp% pgp -ka <filnavn>
    ```
11. *pgp-ca* legger den oppdaterte nøkkelringen ut på UNINETTs web-server. Dette gjøres vhja:
    ```
    pgp% pgp -kx "uninett PGP" pgp-pca.pgp"
    ```
    etterfulgt av en tilsvarende overføring til ftp-området.

10) Pass, førerkort eller norsk bankkort
11) PGP-området er `/usr/home/pgp/cas` (på CA-maskinen)
12) Ftp-området er `/nr/ftp/anonymftp/pub/secude/pgp-pca` (på nettilknyttet server)

F-14 Eksempler på sertifikater

Her følger noen eksempler på utstedte sertifikater under UNISA.

UNINETT CA

Dette er et "standard" CA sertifikat under UNINETT PCA i EuroPKI.

```
Certificate:
SubjectName:            O=Uninett, C=NO
IssuerName:             OU=PCA, O=UNINETT, C=NO
SerialNumber:           4 (decimal)
Validity - NotBefore:   Wed Jan  5 17:33:46 2000 (000105163346Z)
         NotAfter:      Sat Mar 31 23:59:59 2001 (010331215959Z)
Public Key Fingerprint: 9E0A B8E1 E665 35FA 16C0 6BD1 E0C8 F724
SubjectKey:             Algorithm RSA (OID 1.2.840.113549.1.1.1),
                        NULL
          Public modulus (no. of bits = 1024):
              0  F55235EB 6FA54BDE  7EBAD797 5428A63F
             10  751D6198 45F10925  DB6E2B61 3EA7E176
             20  0D498EA1 8302BE2A  E5062AE1 4C1C430E
             30  6F5DA26D 58BBF768  59FC2A57 2C26CEB6
             40  9E78A43C 821F6E72  E27B9BBA 778603E5
             50  6A64637B EA8E11B7  8F2BF1C9 D083837B
             60  9723762B C27EB5D9  A006307B D4DDB31C
             70  C0AD7D27 020BA1F4  6BB1C92A 4274A0B3
          Public exponent (no. of bits = 24):
              0  010001

Certificate extensions:
Authority Key Identifier:00BC 9EA2 5382 1CDB 10A7
                         A8CD 6624 453E 6335 5328
Subject Key Identifier: 55AC AE38 60AF 978B 8A53
                         3E8F EAAE 7BE5 3217 0533
Key Usage:               (CRITICAL) digitalSignature
                         nonRepudiation keyEncipherment
                         dataEncipherment keyCertSign cRLSign
Certificate Policies:    OID 1.3.6.1.4.1.5255.1.1.1
                         OID 1.3.6.1.4.1.2428.10.1.2
Subject alternative names: RFC822: uninett-ca@uninett.no
Subject alternative names: URI: ldap://unisa.nr.no:1522/o=uninett,
                                    c=no?cacertificate
Issuer alternative names: RFC822: pca@uninett.no
Issuer alternative names: URI: ldap://unisa.nr.no:1522/ou=pca,
                                    o=uninett,c=no?cacertificate
Basic Constraints:       allowed to act as a CA !
CRL Distribution Points:
CRL Distribution Point Names: URI: ldap://unisa.nr.no:1522/ou=pca,
                         o=uninett,c=no?certificaterevocationlist

Certificate Fingerprint:8E:D8:AC:A5:97:3E:55:32:
                         55:D5:25:8B:A0:21:6E:38
```

PEM bruker

Dette er et “standard” sertifikat hvor sertifiseringsforespørselen er generert vhja. SECUDE, og sertifikatet vil være installert i SECUDEs PSE.

```
Certificate:
SubjectName:              CN=Anders Lund, <Issuer>
IssuerName:               O=Uninett, C=NO
SerialNumber:             4 (decimal)
Validity  -  NotBefore:   Mon Feb 28 19:47:17 2000 (000228184717Z)
           NotAfter:      Sat Mar 31 23:59:59 2001 (010331215959Z)
Public Key Fingerprint:F2C8 757F 1B1C 21AB 1EAF FE8C 5843 C867
SubjectKey:               Algorithm RSA (OID 1.2.840.113549.1.1.1),
                          NULL
         Public modulus (no. of bits = 512):
            0  15E9A19B 05C62254  20BE2EB4 B4F36ED5
           10  2CFE82E4 3FFBCA26  ED6274CE 7C412341
           20  B3F308FA B870AC51  89C42ABC 2220F0EC
           30  3CC41F88 4A0DC3D7  A84950BA 275A8481
         Public exponent (no. of bits = 24):
            0  010001

Certificate extensions:
Authority Key Identifier: 55AC AE38 60AF 978B 8A53
                          3E8F EAAE 7BE5 3217 0533
Subject Key Identifier:   0EC1 E95E AF90 068D 8B67
                          F159 7BFA AE52 67D8 F3D7
Key Usage:                (CRITICAL) digitalSignature
                          nonRepudiation keyEncipherment
                          dataEncipherment
Certificate Policies:     OID 1.3.6.1.4.1.5255.1.1.1
                          OID 1.3.6.1.4.1.2428.10.1.2
Subject alternative names:RFC822: anders.lund@uninett.no
Subject alternative names:URI: ldap://unisa.nr.no:1522/
                                    cn=anders lund,
                                    o=uninett,c=no?
                                    usercertificate
Issuer alternative names: RFC822: uninett-ca@uninett.no
Issuer alternative names: URI: ldap://unisa.nr.no:1522/o=uninett,
                                    c=no?cacertificate
Basic Constraints:        NOT allowed to act as a CA !
CRL Distribution Points:
CRL Distribution Point Names: URI: ldap://unisa.nr.no:1522/
                                    o=uninett,c=no?
                                    certificaterevocationlist

Certificate Fingerprint:  BB:1C:49:28:5F:67:C0:EE:
                          AA:17:29:0D:50:D3:4C:37
```

Netscape klient

Og dette er et “standard” Netscape klient sertifikat.

```
Certificate:
SubjectName:              CN=Anund Lie, <Issuer>
IssuerName:               O=Norsk Regnesentral, C=NO
SerialNumber:             15 (decimal)
Validity - NotBefore:     Wed Mar 15 17:14:44 2000 (000315161444Z)
         NotAfter:        Sat Mar 31 23:59:59 2001 (010331215959Z)
Public Key Fingerprint:   C720 1489 4EA2 D57C D8A9 916E 9D11 6E62
SubjectKey:               Algorithm RSA (OID 1.2.840.113549.1.1.1),
                          NULL
          Public modulus (no. of bits = 1024):
             0  C847DC12 139DE407  398DB53C 782C8E2D
            10  53CEB228 64EB7B4A  C5C06412 146CDBF2
            20  EDA802BB A827E7A8  E22E1B7D 38BBA86E
            30  EEE98AE2 960CB868  6421D1F3 0742B96E
            40  4969EA12 6CF206D7  1B3F9A5C 55F5A236
            50  A02B2AC4 032C081C  7158064F 3052324A
            60  2ED63474 16BC6A44  E6A7F00F 833A5C79
            70  443A71CE 93AB14B2  07001954 088C93C7
          Public exponent (no. of bits = 24):
             0  010001

Certificate extensions:
Authority Key Identifier: 7835 E3F2 AA90 5894 8B5F
                          425E DB36 035D 3B80 B36B
Subject Key Identifier:   4480 E68D BFC2 67D6 B060
                          B34C DDF8 DE7C A534 DDC7
Key Usage:                (CRITICAL) digitalSignature
                          nonRepudiation keyEncipherment
                          dataEncipherment
Certificate Policies:     OID 1.3.6.1.4.1.5255.1.1.1
                          OID 1.3.6.1.4.1.2428.10.1.2
Subject alternative names: RFC822: anund.lie@nr.no
Subject alternative names: RFC822: anund@nr.no
Subject alternative names: URI: ldap://unisa.nr.no:1522/
                                 cn=anund lie,o=nr,c=no?
                                 usersmimecertificate
Issuer alternative names: RFC822: nr-ca@nr.no
Issuer alternative names: URI: ldap://unisa.nr.no:1522/o=nr,c=no?
                                 cacertificate
Basic Constraints:        NOT allowed to act as a CA !
CRL Distribution Points:
CRL Distribution Point Names: URI: ldap://unisa.nr.no:1522/
                                 o=nr,c=no?
                                 certificaterevocationlist

Certificate Fingerprint:  38:34:58:E7:E0:11:39:97:
                          46:F1:48:5E:F8:75:3F:C2
```

Netscape klient med Netscape extensions

Nedenfor er et sertifikat med Netscape extensions (markert med fet type). Sertifikatet tilhører en PKI forut for EuroPKI. Vi ser også at *Issuer alternative name* og *CRL Distribution Point Names* sine URI angivelser her er *http* og ikke *ldap*.

```
Certificate:
SubjectName:              CN=Espen Haagensen, <Issuer>
IssuerName:               O=Universitetet i Bergen, C=NO
SerialNumber:             2 (decimal)
Validity - NotBefore:     Mon Mar 29 15:24:38 1999 (990329132438Z)
           NotAfter:      Fri Dec 31 13:00:00 1999 (991231120000Z)
Public Key Fingerprint:   F934 94DE DAB9 D1DA A591 26EB A1EC 76DF
SubjectKey:          Algorithm RSA (OID 1.2.840.113549.1.1.1), NULL
         Public modulus (no. of bits = 512):
            0  BF0E0ACD 9B5406B8  CB7CCADC 22AC70B3
           10  624C1E70 B3D7F652  F8D2D7ED 0EA5190E
           20  F95DDA8E A8D1B24D  7B9505ED 39EDCB04
           30  A3B55E22 D410E3DD  541A548F C3CC3865
         Public exponent (no. of bits = 24):
            0  010001

Certificate extensions:
Authority Key Identifier: 1A13 4FA4 DB00 2668 6215
                          1D09 6DF5 8C13 32DC 7F62
Subject Key Identifier:   2F73 FF8C 70CC 2C59 B8F3
                          2287 D7DF C7F0 CCCC AE9D
Key Usage:                (CRITICAL) digitalSignature
                          nonRepudiation keyEncipherment
                          dataEncipherment
Certificate Policies:     OID 1.3.6.1.4.1.2712.10
                          OID 1.3.6.1.4.1.2428.10.1.1
Subject alternative names: RFC822: espenh@ii.uib.no
Issuer alternative names:  RFC822: uib-ca@it.uib.no
Issuer alternative names:  URI: http://www.uninett.no/pca/certs/
                                 uib.cer
Basic Constraints:        NOT allowed to act as a CA !
CRL Distribution Points:
CRL Distribution Point Names: URI: http://www.uninett.no/pca/crls/
                                   uib.crl
Extension OID 2.16.840.1.113730.1.1
         Not critical
         BitString length: 8 bits
                           BitString:
                           0  B0
Extension OID 2.16.840.1.113730.1.13
         Not critical
         IA5String:
           |AUTHENTICATED                  |

Certificate Fingerprint:  92:D2:A6:45:EF:1C:E9:2B:
                          93:FB:A0:29:64:4C:90:4B
```

Web server

Dette er et web-server sertifikat utstedt til Det Medisinske Fakultet ved Universitet i Tromsø. Vi ser at sertifikatet ikke har noe `Subject alternative names:URI: ldap://` … ettersom det er utstedt under en policy om at web-server sertifikater ikke behøver å distribueres.

```
SubjectName:              EMAIL=webmaster@fm.uit.no,
                          CN=www.fm.uit.no,
                          OU=Det Medisinske Fakultet, <Issuer>
IssuerName:               O=Universitetet i Tromsø, C=NO
SerialNumber:             5 (decimal)
Validity - NotBefore:     Wed Oct 11 11:41:01 2000 (001011094101Z)
           NotAfter:      Sat Mar 31 23:59:59 2001 (010331215959Z)
Public Key Fingerprint:   8D5C A6A1 38CD 9C33 DD94 7E4C 6803 354F
SubjectKey:               Algorithm RSA (OID 1.2.840.113549.1.1.1),
                          NULL
          Public modulus (no. of bits = 1024):
             0  D8DA73B1 81A38A9D  B1858B42 6DF8E9D3
            10  8C6F853F 1B0853D3  7D2DB940 78026684
            20  8BAE1DE9 B61579E8  99F7BADC 44B83FB6
            30  170C0FB5 338A80E3  D52CB12B 55628F38
            40  CA95B48A C54D722E  78E1CD24 6130C72B
            50  D4C9A5AA 991A2513  3B5C35E0 8D16E78B
            60  B1E91D7D 7BE6034E  601135B8 A62F1EBC
            70  96A64DC2 B95BF46A  6B383B15 C035D65F
          Public exponent (no. of bits = 24):
            0  010001

Certificate extensions:
Authority Key Identifier: ABE7 7224 CDFA 3973 63D4
                          EA76 42DF 0AC7 84FD 2F01
Subject Key Identifier:   1CD3 5976 8626 4CDE 6DD2
                          43B7 9323 DBD1 0233 61C2
Key Usage:                (CRITICAL) digitalSignature
                          nonRepudiation keyEncipherment
                          dataEncipherment
Certificate Policies:     OID 1.3.6.1.4.1.5255.1.1.1
                          OID 1.3.6.1.4.1.2428.10.1.2
Subject alternative names: RFC822: webmaster@fm.uit.no
Issuer alternative names:  RFC822: uit-ca@cc.uit.no
Issuer alternative names:  URI: ldap://unisa.nr.no:1522/o=uit,
                           c=no?cacertificate
Basic Constraints:        NOT allowed to act as a CA !
CRL Distribution Points:
CRL Distribution Point Names: URI: ldap://unisa.nr.no:1522/o=uit,
                          c=no?certificaterevocationlist

Certificate Fingerprint:  FE:42:1B:3F:7B:00:17:7D:
                          50:31:B0:43:B4:7E:A0:F8
```