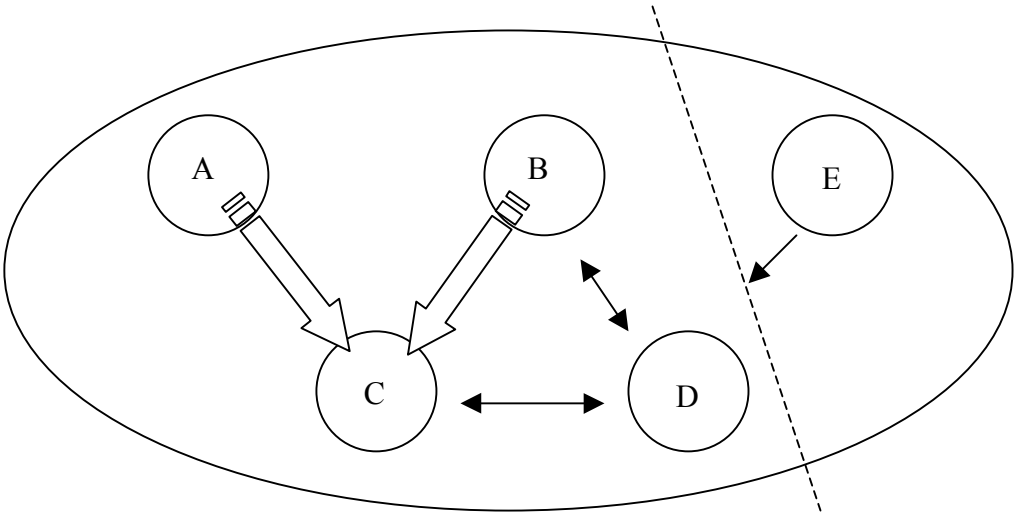
FOI
TOTALFÖRSVARETS
FORSKNINGSINSTITUT

FOI-R--0695--SE
November 2002
ISSN 1650-1942
Användarrapport

Alf Bengtsson, Amund Hunstad, Lars Westerdahl

Autonomitet vid autentisering i nätverksbaserade system



Ledningssystem
581 11 Linköping

TOTALFÖRSVARETS FORSKNINGSINSTITUT

Ledningssystem
Box 1165
581 11 Linköping

FOI-R--0695--SE

November 2002

ISSN 1650-1942

Användarrapport

Alf Bengtsson, Amund Hunstad, Lars Westerdahl

Autonomitet vid autentisering i nätverksbaserade system

Utgivare	Rapportnummer, ISRN	Klassificering
Totalförsvarets Forskningsinstitut - FOI Ledningssystem Box 1165 581 11 Linköping	FOI-R--0695--SE	Användarrapport
	Forskningsområde 4. Spaning och ledning	
	Månad, år November 2002	**Projektnummer** E7023
	Verksamhetsgren 5. Uppdragsfinansierad verksamhet	
	Delområde 41 Ledning med samband och telekom och IT-system	

Författare/redaktör	Projektledare
Alf Bengtsson Amund Hunstad Lars Westerdahl	Alf Bengtsson
	Godkänd av Lennart Nyström
	Uppdragsgivare/kundbeteckning FM
	Tekniskt och/eller vetenskapligt ansvarig Alf Bengtsson

Rapportens titel

Autonomitet vid autentisering i nätverksbaserade system

Sammanfattning (högst 200 ord)

Visionen om ett nätverksbaserat försvar medför vissa krav på det kommande ledningssystemet. Ett viktigt krav är förmågan till autonomitet. I rapporten presenteras fyra aspekter på autonomitet - globalt beroende, förutsedd autonomitet, oförutsedd autonomitet under egen kontroll respektive utom egen kontroll. Utifrån dessa aspekter diskuteras tre olika klasser av autentiseringsmetoder; biljett-, certifikat- respektive identitetsbaserade metoder. Med autentisering avses äkthetsbekräftelse av angiven identitet med hjälp av digitala metoder. Analoga metoder, t ex biometrisk autentisering, diskuteras inte.

Rapporten avslutas med en värdering av de tre autentiseringsklasserna med avseende på autonomitet. De olika metoderna möter kraven i olika grad. Slutsatsen är att certifikatbaserade system är mest generellt användbara, särskilt som de kan kompletteras för att direkt understödja autonomoitet. Biljettbaserade metoder kan hantera förutsedd autonomitet. Identitetsbaserade metoder är alltför oflexibla för att kunna användas generellt.

Nyckelord

autentisering, IT-säkerhet, PKI, autonomitet, nätverksbaserat försvar

Övriga bibliografiska uppgifter	Språk Svenska
ISSN 1650-1942	**Antal sidor:** 47 s.
Distribution enligt missiv	**Pris: Enligt prislista**

Issuing organization
FOI – Swedish Defence Research Agency
Command and Control Systems
P.O. Box 1165
SE-581 11 Linköping

Report number, ISRN
FOI-R--0695--SE

Report type
User report

Research area code
4. C4ISR

Month year
November 2002

Project no.
E7023

Customers code
5. Commissioned Research

Sub area code
41 C4I

Author/s (editor/s)
Alf Bengtsson
Amund Hunstad
Lars Westerdahl

Project manager
Alf Bengtsson

Approved by
Lennart Nyström

Sponsoring agency
Swedish Defence

Scientifically and technically responsible
Alf Bengtsson

Report title (In translation)
Autonomy for Authentication in Network Based Systems

Abstract (not more than 200 words)
The vision of a network based defence involves a set of demands for the C^2 system. One important requirement is the ability to act autonomously. In this report we present four aspects of autonomy - global dependencies, predictable autonomity, autonomity not predictable but controlled and not controlled respectively. Three classes of methods for authentication - ticket, certificate and identity based respectively - are discussed in relation to these four aspects. With authentication we mean verification of claimed identity by digital means. We don't discuss analog methods, e. g. biometric authentication.

We present a table of assessments for the three classes of authentication related to the four aspects of autonomy. The three classes meet the demands in different ways. Our conclusion is that methods based on certificates are most applicable. They can also be enhanced by other methods to support autonomy. Ticket based methods can be used when the autonomy is predictable. Identity based methods are too inflexible for general use.

Keywords
authentication, IT-security, PKI, autonomy, network centric defense

Further bibliographic information

Language Swedish

ISSN 1650-1942

Pages 47 p.

Price acc. to pricelist

Innehåll

Figur

Tabell

1 Inledning

Autentisering, dvs. verifiering av identiteten hos en användare eller annan aktör, kan utföras enligt olika metoder. I en tidigare rapport [BEN01] jämförs tre klasser av metoder, utifrån en uppsättning krav som ställs av de uppgifter som skall lösas av det framtida nätverksbaserade försvaret. Olika krav löses olika bra av de tre klasserna, varför en entydig värdering inte presenterades. Sammantaget förordades dock biljettbaserade respektive certifikatbaserade metoder framför den tredje klassen, identitetsbaserade metoder.

Ett av de viktigaste kraven är förmågan att kunna agera autonomt. Därför är detta krav djupare analyserat i föreliggande rapport. Autonomitet krävs i olika avseenden. T.ex. skall enheter kunna agera autonomt om de oplanerat blir avskurna från resten av systemet. Ett annat krav är att man (snabbt) skall kunna sätta ihop olika typer av förband och enheter som självständigt skall kunna agera för att lösa en uppgift.

Man kan dela in autentiseringsmetoder i tre huvudklasser; biljett-, certifikat- och identitetsbaserade metoder. I denna rapport presenteras dessa klasser med några exempel ur varje klass. De olika klasserna analyseras med avseende på deras förmåga att tillgodose olika aspekter av autonomitet.

De slutsatser som dras i rapporten är sammanfattade i kapitel 2. I kapitel 3 diskuteras den vision som ligger till grund för ett nätverksbaserat försvar. Utifrån detta formuleras några krav på ledningssystemet. Speciellt beskrivs i 3.2 fyra aspekter på autonomitet - globalt beroende, förutsedd autonomitet, oförutsedd autonomitet under egen kontroll respektive utom egen kontroll. Dessa återkommer sedan vid analysen av de tre klasserna autentiseringsmetoder. Kapitel 4 beskriver fem relevanta frågeställningar på autentisering. Tre olika autentiseringsklasser presenteras i kapitel 5 och jämförs i kapitel 6, där också slutsatserna dras.

2 Sammanfattande slutsatser

Tre klasser av autentiseringsmetoder jämförs - biljettbaserade metoder, certifikatbaserade metoder respektive identitetsbaserade metoder. De värderas utifrån deras möjlighet att understödja olika typer av autonomitet. Främst analyseras autonomitet som varit förutsedd redan vid design av systemet, kontra olika grader av oförutsedd autonomitet.

Biljettbaserade metoder är beroende av frekvent kontakt med centrala tjänster. Förutsedd autonomitet kan hanteras. Däremot är det hög risk för misslyckande vid oförutsedd autonomitet.

Certifikatbaserade metoder har bäst förmåga att hantera autonomitet. De kan också kompletteras med metoder som ytterligare ökar förmågan till autonomitet. Bland de certifikatbaserade metoder som analyserats bedöms SPKI/SDSI vara mest flexibla.

Identitetsbaserade metoder är alltför oflexibla för att kunna användas generellt. De kan bara komma ifråga inom lokala grupper som varit förutsedda redan vid design av systemet.

3 Relation till visionen om ett nätverksbaserat försvar

I en tidigare rapport [BEN01] beskrivs autentisering i nätverksbaserade system. Vi utgick då från en lista av krav som vi identifierade ur försvarsmaktens vision om ett nätverksbaserat försvar. Eftersom det är denna vision som är vägledande även i föreliggande rapport återges här ännu en gång merparten av de identifierade kraven.

3.1 Identifierade krav

Kraven på det framtida informations- och ledningssystemet inom försvaret finns inte formulerade i detalj. Detta är naturligt, eftersom en grundtanke är att systemet kontinuerligt skall kunna modifieras i takt med teknikutvecklingen och i takt med att försvarets uppgifter växlar. Detta ger i sig självt ett ytterst väsentligt krav – systemet kan inte vara ett stort, monolitiskt system. Det måste bestå av komponenter och delar, som kopplas ihop via standardiserade gränssnitt, så att det går att modifiera en komponent utan att hela systemet påverkas.

Grundkravet på informations- och ledningssystemet är att det skall användas för att effektivt leda förband i väpnad strid. Det är därför viktigt att systemet följer den doktrin som försvarsmakten har för ledning. Det dokument där denna doktrin sammanfattas är "Försvarsmaktens Grundsyn Ledning" [FM01]. Några citat ur denna kan ge riktlinjer.

"Chefen är ytterst ansvarig för uppgiftens lösande och de beslut som fattas. Detta ansvar kan inte delegeras. Befogenheter tilldelas alltid i paritet med ansvar. Enkla och tydliga ansvars- och lydnadsförhållanden skall eftersträvas."

"Det militära försvarets ledningsmetod är uppdragstaktik."

"Förband i insatsorganisationen skall ha sådan förmåga till självständigt uppträdande att de kan agera i enlighet med överordnad chefs intentioner även om förbindelsen med denne brutits."

"Den framtida striden kommer att ställa allt högre krav på att rätt verkan sätts in, på rätt plats och i rätt tid. Försvarsmaktens vision och strävan är därför utveckling mot så kallad nätverkscentrerad krigföring, med kraftigt förbättrade möjligheter till samordning."

"Chef för insatsstyrka kan tilldelas operativt eller taktiskt ledningsansvar. Vid multinationella insatser kan, med särskilda begränsningar, svenska förband lyda under utländsk chef."

Termen "nätverkscentrerat försvar" är inte närmare definierad i [FM99]. Den innebörd vi lägger i termen är bland annat att det inte skall finnas organisatoriska eller tekniska murar som stänger informationsflödet mellan två aktörer som är behöriga till informationen för att lösa en beordrad uppgift. Tekniken skall möjliggöra informationsflöden såväl i en hierarkisk struktur som i en flatare struktur. Kommunikationsfunktionerna skall vara integrerade i ledningssystemet.

Förutom grundkravet, att ledningssystemet skall användas för ledning av väpnad strid, tillkommer flera andra krav. Ledningssystemet skall stödja försvarets logistik, det skall kunna samverka med civila system m m.

Försvarets uppgifter skall dimensionera systemet. Vilka uppgifter som är aktuella att lösa om 10-20 år går förstås inte att förutspå idag. Men i andra planeringssammanhang bygger man upp målbilder och scenarier baserade på olika kombinationer av de fyra huvuduppgifter som anges idag, enligt Försvarsplan 2000 [FM00]. Det är därför rimligt att fundera över dessa och försöka bedöma respektive huvuduppgifts viktigaste konsekvenser för ledningssystemet.

- VA, Väpnat Angrepp. Den mest tekniktunga uppgiften. I händelse av ett väpnat angrepp skall angriparen kunna mötas över hela det operativa djupet, dvs. Sveriges hela territorium – mark, sjö och luft. Insatsstyrkor, med ledning från rörliga insatsstaber, skall kunna sättas samman av enheter ur alla försvarsgrenar. Sensorer och plattformar skall kunna avläsas och styras på avstånd från de rörliga staberna. För att snabba upp beslutsprocessen skall information kunna överföras mellan olika ledningsnivåer och förband.

  Viktiga konsekvenser av Väpnat Angrepp:

  Kommunikation med hög kapacitet över hela territoriet. Stora mängder information från många olika källor skall vara tillgänglig, för alla som är behöriga, oberoende av förbandsstruktur. Sensor- och plattformsstyrning, skall vara möjlig från ledningsstab. "Små ledningssystem" för insatsstyrkor måste snabbt kunna sättas ihop.

- TI, Territoriell Integritet. Ett skalskydd runt Sverige. Färre sensorer och plattformar och lägre krav på kommunikation över djupet än i uppgift Väpnat Angrepp.

  Inga tillkommande konsekvenser jämfört med Väpnat Angrepp.

- II, Internationella Insatser. Försvaret skall kunna bidra till stabiliserande och krisdämpande internationella insatser. Internationell samverkan ställer stora krav på interoperabilitet. Insatsstyrkorna sätts samman av enheter ur flera försvarsgrenar. Ledningen består av central stab, som kan vara placerad i Sverige, samt operativ insatsstab, med rörliga delar, som kan ingå i multinationella stabskonstellationer.

  Viktiga konsekvenser av Internationella Insatser:

  Ett "litet ledningssystem" måste snabbt kunna sättas ihop, liksom i uppgift Väpnat Angrepp. Det som tillkommer är att det skall kunna verka på främmande territorium och med långdistansförbindelse med Sverige. Den mest framträdande konsekvensen av Internationella Insatser är att information skall kunna utväxlas med andra nationers ledningssystem. Detta skall ske på ett så effektivt sätt att samverkansoperationer kan genomföras.

- SS, Stöd till Samhället. Försvarsmakten skall kunna stödja samhället vid t.ex. katastrofer, omfattande terrorism eller grov internationell brottslighet.

  Den mest påtagliga konsekvensen av Stöd till Samhället är kraven på samverkan med det civila samhällets informationssystem. Dessa krav finns också i övriga uppgifter.

Ovanstående axplock ur beskrivningar av försvarsuppgifter och visioner medför ett antal krav på ledningssystemet; krav som är delvis motstridiga. Dessa sammanfattas i följande punktlista. Listan är förstås inte fullständig, men kan likväl tjäna som en kravlista. Som ett allomfattande krav på det framtida ledningssystemet finns "systemet skall vara tillräckligt säkert och robust". I denna rapport avgränsas aspekter på säkerheten till enbart autentiseringsfunktionen.

A. Civil teknik och civila system måste användas i högsta möjliga grad. Detta är inte enbart av kostnadsskäl, utan också en konsekvens av uppgifterna. Detta innebär i praktiken att grunden till systemet är framtida Internetteknik, som på något sätt måste göras "tillräckligt säker". Flexibilitetskrav gör att man skall välja leverantörs- och plattformsoberoende lösningar.

B. Den primära uppgiften hos ledningssystemet är att göra information tillgänglig; på rätt plats, i rätt tid och för rätt aktör. Som sekundär uppgift finns att, under vissa betingelser, möjliggöra styrning av sensorer och plattformar. Informationen, och styrkommandona, kan vara av olika slag – lägesbilder, order, mobil kod mm.

C. Verksamheten är organiserad i en befäls- och ansvarshierarki. Tekniken i ledningssystemet skall ändock underlätta informationsutbyte oberoende av nivågränser.

D. Informationen skall finnas så nära ägaren (insamlaren) som möjligt. Detta bl.a. för att tydliggöra ansvarsförhållanden.

E. Delar av systemet skall kunna knoppas av, eller på annat sätt sättas upp, vid uppbyggnad av insatsstyrkor, eller vid andra behov av autonoma delsystem.

F. System av system. Bland annat kraven på interoperabilitet och utbyggbarhet gör att det totala systemet måste bestå av delsystem som är hopkopplade via standardiserade gränssnitt.

3.2 Autonomitet

Syftet med föreliggande rapport är att närmare granska kravet E ovan, möjligheten att skapa autonoma delsystem. Nationalencyklopedins definition av *autonom* är *självständig, oberoende*.

Ett datorsystem består av flera delsystem. Dessa system samverkar för att lösa en gemensam uppgift. För att möjliggöra att rätt personer och tjänster (i det följande är personer, tjänster, datorer m m sammanfattade i begreppet *aktör*) har access till varandra krävs autentisering av dessa enheter. Autentiseringen medför att man kan verifiera identiteten på aktören. Detta är det nödvändiga första steget för att kunna kontrollera vilka rättigheter aktören har att utnyttja systemet.

I ett nätverksbaserat försvar kommer ett antal delsystem att samverka för att lösa försvarsmaktens fyra huvuduppgifter. Delsystemen kommer att vara spridda på ett flertal plattformar, med delvis olika förutsättningar att skicka, ta emot och behandla information. Autentisering är en grundförutsättning för att kunna hantera informationsutbytet. Men alla plattformar, och andra delar av det totala systemet, kan inte i varje givet ögonblick ha kontakt med varandra. Detta medför att ett autentiseringssystem måste klara av ett visst mått av autonomitet. Vi väljer att

diskutera graden av autonomitet ur fyra aspekter - beroendet av *globala tjänster*, möjlighet till *förutsedd autonomitet*, hantering av *oförutsedd men styrd autonomitet* samt *oförutsedd, icke styrd autonomitet*.

I. *Beroende av globala tjänster*. Autentisering handlar ytterst om förtroende. Flera autentiseringssystem har en hierarkisk struktur där en toppdomän är den alla litar på, medan underdomänerna kan vara misstänksamma mot varandra. Beroendet av en toppdomän kan medföra att hela autentiseringssystemet fallerar om kontakten med toppdomänen bryts eller störs ut. System utan hierarki är naturligtvis mindre känsliga, men medför att mycket av autentiseringsarbetet förs över på användarna och blir därigenom mer godtyckligt.

   Det finns också andra komponenter i ett system som stödjer funktionaliteten och säkerheten. Ett exempel på detta är ett, globalt, gemensamt tidssystem. Tiden kan användas som en parameter för att undvika att samma meddelande återanvänds. Tidsdifferensen mellan olika händelser kan indikera olika grad av tilltro. Om tidsdifferensen mellan två händelser är för stor, eller för liten, kan det medföra att händelserna förkastas.

   Beroendet av globala tjänster är ett övergripande problem inom ett nätverksbaserat system och bör i möjligaste mån minimeras.

II. *Förutsedd autonomitet*. Vid konstruktionen av ett större nätverk delas helheten upp i mindre segment. Syftet är att göra hela nätverket gripbart och att upprätthålla prestanda i nätet. Uppdelningen kan, med avseende på autentisering, beskrivas som att man gör ett stort problem till flera mindre. Naturligtvis är det enklare att hantera förtroendefrågor i ett mindre sammanhang. Dessutom kan det vara möjligt för segmentet att fungera på egen hand, i händelse att kontakten med övriga segment bryts.

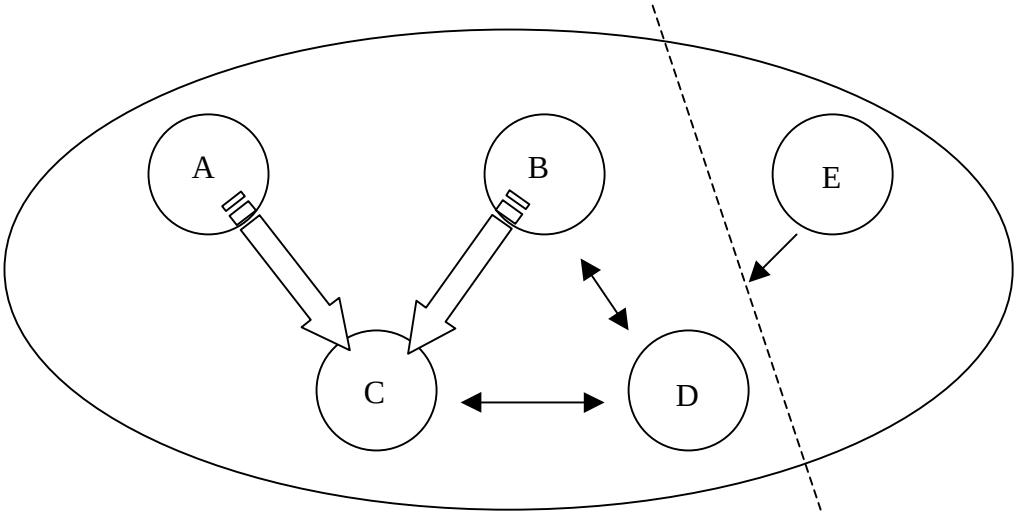
   Det sker inte bara en fysisk uppdelning av ett nätverk. Tjänster fördelas och görs tillgängliga beroende på vilka delar som har behov av dessa tjänster. Även tjänster kan delas upp i huvudtjänster och understödjande tjänster, vilket medför olika beroendegrader beroende på typ av tjänst.

III. *Oförutsedd men styrd autonomitet*. Tanken med ett nätverksbaserat försvar är att delar av olika system skall kunna samverka på en mer direkt nivå än vad som är möjligt i nuläget. Ett system skall beredas teknisk möjlighet att få tillgång till all information och kommunikationsförmåga som krävs för att lösa dess uppgift. Det innebär bl a att en tillfällig enhet skall kunna sättas upp för att lösa en specifik uppgift, utan att detta varit förutsett vid designen av systemet. Denna tillfälliga enhet kan bestå av komponenter från flera olika delsystem, vilket medför att de måste ha möjlighet att skapa ett gemensamt förtroende, dvs. kunna autentisera sig mot varandra utan att vara beroende av att utnyttja högre, globala system.

IV. *Oförutsedd, icke styrd autonomitet*. Det kan hända att ett system förlorar kontakten med övriga system. Orsakerna kan vara fysiska, dvs. att kontakten bryts på grund av en tråd går av eller att en enhet hamnar i radioskugga. Även om nätet är fysiskt intakt, kan en tjänst likväl tillfälligt slås ut. Ett robust system kan fortfarande fungera lokalt även om kontakt med övriga delar tillfälligt saknas.

De två första nivåerna är av mer övergipande karaktär, då de tar upp generella problem samt problem med system arkitektur. Den styrda autonomiteten är en verklig utmaning för det nätverksbaserade försvaret. Här i ligger svårigheterna att få flera delsystem att kommunicera med varandra samt att utbyta information på ett meningsfullt och funktionellt sätt.

Figur 3.1 beskriver ett scenario från ovanstående lista.



Figur 3.1. *Scenario*. A och B skapar C med delar ur sig själva för att lösa en viss uppgift. Både B och C kan tala med D. E har blivit avskuren från hela systemet.

4 Autentisering, ur perspektivet autonomitet

Autentisering betyder äkthetsbekräftelse. Metoder för autentisering betyder alltså metoder att verifiera att någonting är äkta. I [BEN01] beskrivs autentisering allmänt i distribuerade system. I föreliggande rapport fokuseras på de delar som har inverkan på autonomitet.

Med autentisering menar vi äkthetsbekräftelse av identitet. En aktiv aktör skall kunna verifiera att den identitet som en annan aktör uppger inte är falsk. Vidare avgränsas till "digitaliserade aktörer", t.ex. sensorer, datorprogram, datorer och kommunikationsutrustning. Matchning av fingeravtryck och andra "analoga metoder" att identifiera människor avhandlas alltså inte. Däremot autentiseras även människor ofta via "digitaliserade aktörer", t.ex. aktiva kort.

Digital autentisering tillgår så att en aktör styrker sin identitet genom att bevisa att han besitter en hemlighet, autentiseringsnyckeln. Han använder denna nyckel till att räkna fram någon form av data som presenteras för den andre aktören. Denne har tillgång till en verifieringsnyckel, som han på något sätt vet är knuten till motpartens uppgivna identitet. Han kan med hjälp av verifieringsnyckeln verifiera att de data han mottagit måste ha beräknats med hjälp av rätt autentiseringsnyckel.

Ett oundvikligt steg i alla autentiseringssystem är att objekt med tillhörande identitet måste tillföras systemet, de måste "födas" (och också "dö"). För att man skall kunna ha tilltro till säkerheten måste födelsen övervakas, och styrkas, av någon betrodd aktör, t.ex. en administratör (eller aktörer i samverkan). Ett exempel är att i vissa system tillåts inte aktörerna att själva skapa sina nycklar. En administratör måste skapa nycklarna, och gå i god för att de uppfyller nödvändiga krav. Nycklarna måste sedan på ett säkert sätt överföras till aktören. En viktig fråga är om administratörsrollen kan distribueras till flera administratörer.

För att autentiseringen skall vara pålitligt säker krävs också att den beräkning, där autentiseringsnyckeln ingår, skall vara kryptologiskt stark. Det skall vara en så stark envägsfunktion att det tar "orimligt lång tid" att baklänges, utifrån presenterade resultatdata, räkna ut eller leta fram den använda autentiseringsnyckeln. I annat fall kan en obehörig räkna fram autentiseringsnyckeln och sedan maskera sig under falsk identitet. Kryptologiska aspekter behandlas inte här utan det förutsättes vara gjort av experter på kryptologi. I de flesta autentiseringssystem kan man välja autentiseringsalgoritm beroende på hur starka krav man har.

Däremot diskuteras andra frågeställningar, som har inverkan på autonomitet, i denna rapport

(a) Hur visar man, på ett säkert sätt, att man besitter autentiseringsnyckeln? Detta är bl.a. de beräkningar med autentiserings- och verifieringsnyckeln som nämndes ovan, samt överföringen av beräkningsresultaten. För att t.ex. förhindra återanvändning av gamla beräkningsresultat skall t.ex. slumpade data och/eller tidsstämplingar ingå. Hela detta steg kallas autentiseringsprotokoll, se vidare [SCH96a].

(b) Hur knyter man, på ett säkert sätt, ihop en aktörs verifieringsnyckel med hans identitet (annars kan han uppträda under falskt namn)?

(c) Hur löses problemet att en aktör skall kunna agera under olika identiteter (t.ex. i olika roller eller genom olika namngivningssätt i olika delsystem)

(d) Hur hanteras födelseprocessen? Kan denna delegeras, så att autonoma system kan sättas samman?

(e) Hur hanteras dödsprocessen, spärrlistor och andra sätt att spärra ut data (identiteter, nycklar mm) som inte längre är pålitliga.

Alla dessa frågeställningar är väsentliga att lösa i det komplexa system-av-system som är grunden i det nätverksbaserade försvaret. När det gäller förmågan till autonomitet är punkterna (d) och (e), som handlar om administrationen av objekt och deras identiteter, av speciell vikt. Dessa förtjänar därför en speciell diskussion, se följande kapitel 4.1.

4.1 Introducering och avveckling i ett system

En förutsättning för att en aktör - person, maskin, datorprogram eller tjänst - skall kunna autentiseras är att systemet känner till aktörens korrekta identitet. Ett system kan inte verifiera en aktör utan att en inmatningsprocess, där aktören introduceras i systemet, har genomgåtts.

Informationen från en inmatning måste också förmedlas till alla instanser som behöver informationen för att kunna autentisera aktören. I centralt uppbyggda system är detta enklare då all information hämtas från samma, eller i alla fall en begränsad mängd, instanser. Med decentraliserade system sprids informationen och förmågorna över flera instanser, vilket komplicerar säkerheten vid distribution av informationen till alla behövande.

Problematiken är likartad när det gäller att avveckla en aktör ur systemet. Även i detta fall är det en viss mängd information som måste nå alla kritiska instanser. En skillnad gentemot introducering är att vid avveckling är det ofta så att aktören inte kan, eller vill, medverka vid informationsspridningen.

Oavsett vilken metod eller modell som används i ett system för autentisering är introduceringen och avvecklingen av personer, maskiner och tjänster kritisk för den övergripande säkerheten. Kan man inte styrka identiteten på den som skall introduceras respektive avvecklas kommer identiteten aldrig att kunna verifieras korrekt och därmed säkert.

4.1.1 Att introduceras i ett system

Att introduceras, eller om man så vill att födas, i ett system är en förutsättning för att systemet skall känna till en aktörs existens. Aktören måste då kunna visa upp någon form av medlemskap eller rättighet att få nyttja systemet. I ett litet och slutet system är detta inte speciellt komplicerat då alla nyttjare är överskådliga. Problemen växer i takt med systemets storlek och spridning. I ett globalt system är det opraktiskt, för att inte säga omöjligt, att centralt sköta all introducering. Om man tar Internet som exempel så är detta inte ett stort globalt system, utan en sammanslutning av flera mindre system. I alla dessa mindre system har varje aktör introducerats med hjälp av någon metod, men det betyder inte att andra

system har godkänt denne aktör eller för den delen vet vem det är. En kontrast till Internet är ett avgränsat, militärt system där de ingående aktörerna är väl kända.

Förutsättningarna för att introduceras i ovanstående system är mycket olika, så även behovet av autentisering. När ett system skapas görs en avvägning mellan säkerhet och bekvämlighet [SMI01]. I uttrycket bekvämlighet innefattas mängden aktörer samt spridningen av dessa.

[SMI01] delar in introduceringen i självautentisering och personlig autentisering. I större system, som är allmänt tillgängliga och mindre kritiska, är självautentisering vanlig. Det kan innebära att en användare på egen hand skapar en identitet, t.ex. genom att ange ett användarnamn och ett lösenord. Varianter finns där användaren får ett givet lösenord bara för att systemet skall fungera och att sedan användaren själv ändrar detta. Riskerna med dessa varianter är att användaren skapar dåliga lösenord alternativt glömmer att ändra på det givna lösenordet.

En säkrare, men mer krävande, metod är när aktören, eller någon person som ansvarar för den maskin eller tjänst som skall introduceras, personligen måste närvara vid introduceringen. I vissa fall är blotta närvaron en styrkning av personens rättighet att nyttja systemet. Utöver id-handlingar kan identiteten och nyttjanderätten styrkas av en medföljande person eller anställningsbevis, kreditivbrev etc.

Introducering med lösenord används för mänskliga aktörer. Vi har i denna rapport emellertid huvudsakligen inriktat oss på digitala aktörer. Som tidigare nämnts knyts ett nyckelpar till sådana digitala aktörer. En autentiseringsnyckel, som skall hållas hemlig, och en verifieringsnyckel, som skall spridas. Introduceringen av en digital aktör kan ofta delas upp i två steg. Ett första steg innebär skapande av aktörens nyckelpar. Ett andra steg är att aktörens identitet, på ett säkert och verifierbart sätt, skall knytas till verifieringsnyckeln.

Ibland önskar man att autentisera en människa med hjälp av en digital aktör, t.ex. ett aktivt kort. Men då måste man säkerställa, t.ex. via biometrisk autentisering, att enbart rätt person tillåts använda kortet. Detta problem avhandlas inte vidare här.

Det första steget vid introduceringen, skapande av nyckelparet, kan göras på två sätt. I det enklaste fallet genererar aktören själv sitt nyckelpar och skickar sin publika verifieringsnyckel i ett signerat meddelande till en administratör, som då kan bekräfta att aktören använt rätt hemlig autentiseringsnyckel vid signering av meddelandet. Ett annat alternativ är att administratören genererar nyckelparet, men då tillkommer problematiken med hur den hemliga nyckeln skall transporteras till dess ägare. Som vi beskrivit detta kräver inget av alternativen att aktören är fysiskt närvarande. Men det är mycket svårt att få en säker introducering om inte aktören i något skede är fysiskt närvarande. Enda lösningen är via ett eller flera betrodda ombud (jfr. delning nedan).

För det andra steget, hopknytning av identitet och verifieringsnyckel, används i ett centraliserat system någon form av register och i ett decentraliserat system ofta någon variant av digitala certifikat. Ett digitalt nyckelcertifikat är en bekräftelse på att en publik nyckel till hör en specifik identitet. Denna bekräftelse är digitalt signerad med hjälp av certifikatutfärdarens autentiseringsnyckel. Alla aktörer som litar på utfärdaren, och som har tillgång till utfärdarens verifieringsnyckel, kan verifiera att certifikatet är äkta.

4.1.2 Att avvecklas ur ett system

Lika viktigt som det är att introduceras korrekt och säkert i ett system är det att kunna avvecklas ur samma system. Orsaken till att identiteten av en person, maskin eller tjänst skall försvinna ur systemet kan variera. En person kan sluta på sin tjänst och därmed inte längre ingå i en given organisation. Även om personen inte slutar men får nya arbetsuppgifter kan rättigheter och organisationstillhörighet ändras på ett sådant sätt att exempelvis ett nytt certifikat behöver utfärdas. Om identiteten tillhör en maskin eller tjänst kan det hända att dessa skall tas ur drift eller modifieras så att nya förutsättningar gäller för nyttjandet.

Andra oförutsedda och mer oroväckande orsaker kan vara att ägarens hemliga nyckel exponeras, genom intrång eller oförsiktighet, och därmed äventyrar systemidentiteten. Hur detta upptäcks ligger utanför denna rapports område.

Oavsett orsak till att en identitet skall avvecklas finns det problem med hur detta skall säkerställas. Likt introduceringen står även här problemen i proportion till storleken av systemet, men även vilken autentiseringsmodell som används. Det man främst skiljer på är om systemet är centraliserat eller decentraliserat.

Ett centraliserat system (serverbaserat) har kortare ledtider mellan beslut och verkan. Från det att beslutet fattas att en identitet skall avvecklas går det fort att genomföra detta i praktiken, då autentiseringen sker centralt. Avvecklingen genomförs genom att identiteten markeras som ”avvecklad” i den lista över identiteter som den centrala servern har. Alternativt kan servern skapa en eller flera listor (jfr. spärrlistor) med avvecklade identiteter.

I decentraliserade system förlängs denna period i och med att informationen måste spridas längre och att alla måste uppdatera sin information. Informationen sprids i form av meddelanden på samma sätt som certifikaten vid introduceringen. Avgörande för snabbheten i ett decentraliserat system är hur informationen sprids. Ett problem är att avvecklade aktörer inte själva kan, eller vill, medverka i spridningen av informationen. Detta är en väsentlig skillnad gentemot introduceringen där aktören har allt intresse av att sprida sitt certifikat.

4.1.3 Administratörsberoende

I beskrivningen av introducering och avveckling är det underförstått att det finns en betrodd administratör som hanterar vart och ett av de fyra kritiska momenten – identifiering plus nyckelgenerering, centralt register eller certifikatutfärdande, beslut om avveckling respektive utfärdande av spärrinformation. Det är inte nödvändigtvis samma administratör som hanterar alla fyra momenten.

Om ett visst moment administreras av en enda administratör blir detta förstås en sårbar punkt. Det är heller inte bra för autonomiteten i och med att en aktör kan ha svårt att komma i kontakt med den enda administratören. Den naturliga ansatsen är att tillåta flera administratörer av ett moment, t.ex. att både administratör A och administratör B är behöriga att signera certifikat. Men denna redundans fås på bekostnad av säkerheten. Motsidan får ju flera alternativa administratörer att kompromettera och därmed större möjlighet att skapa falsk information. Dessutom riskerar man att få in motstridig information i systemet om man har flera administratörer för samma moment.

Ett intressant sätt att minska sårbarheten är "administration via ombud". Tanken är att administratören tar hjälp av ett antal "vanliga aktörer" som via en extra kontroll på något sätt befunnits vara extra betrodda och kompetenta och som därför har valts ut till ombud. Eftersom ett ombud inte är lika betrott som "den riktige administratören" krävs ofta att mer än ett ombud involveras. I en (k,n)-kombination finns n st ombud utsedda. För att ett administrativt beslut skall vara giltigt krävs att k st av ombuden involveras. Om det är så att en aktör har lättare att komma i kontakt med k ombud än med den enda administratören har bl a autonomiteten underlättats. Likaså har sårbarheten minskats, om n>>k, i och med att det finns n st alternativa ombud att välja bland.

5 Tre klasser av autentiseringsmetoder

Detta avsnitt handlar om vissa egenskaper hos tre olika klasser av autentiseringsmetoder, och hur frågeställningarna a)-e) i kap 4 löses. Klasserna kommenteras sedan i relation till aspekterna I-IV, de olika aspekterna på autonomitet i kap 3.

5.1 Biljettbaserad autentisering: Kerberos

I [BEN01] beskrevs i huvudsak Kerberos, men även KryptoKnight och DCE, som exempel på biljettbaserade system. Både KryptoKnight och DCE är system vilka bygger på Kerberos-teknik, och är därmed förhållandevis lika i funktion och beteende.

Biljettbaserade, ofta också kallade serverbaserade, system är i huvudsak system, som med hjälp av en central server autentiserar aktörer. En aktör måste inledningsvis logga in på en server vilken har möjlighet att verifiera om aktören är en giltig användare av systemet som helhet. Därefter begär aktören kortvariga biljetter (sessionsnycklar) av en biljettserver (TGS) till de tjänster som är av intresse.

Kerberos delas in i domäner, vanligtvis kallade realm:s. Varje realm kontrollerar ett antal aktörer och tjänster. Vill en aktör eller tjänst utnyttja en tjänst i en annan domän går detta bra i och med att autentiseringsservrarna skall ha utbytt nycklar med varandra vid initieringen.

Hjärtat i en realm är nyckelservern (KDC:n). Om en aktör inte kan etablera en kontakt med KDC:n är det omöjligt att autentisera sig och därmed få tillgång till de tillfälliga nycklar som används för att nå tillgängliga tjänster.

Utvecklar man autentiseringskonceptet enligt kapitel 4 har Kerberos följande egenskaper;

(a) Hur visar man, på ett säkert sätt, att man besitter autentiseringsnyckeln? Varje aktör måste initialt ansluta sig till systemet med ett lösenord. Detta lösenord skickas aldrig, i någon form, över nät. KDC sänder aktören en sessionsnyckel till TGS:en, krypterad med användarens lösenord [PFL97]. Om användaren kan dekryptera sessionsnyckeln korrekt är denna möjlig att använda för vidare kommunikation.

(b) Hur knyter man, på ett säkert sätt, ihop en aktörs verifieringsnyckel med hans identitet? Kerberos och KryptoKnight använder sig av symmetriska nycklar, dvs. signerings- och verifieringsnyckeln är den samma. Det är fullt möjligt att implementera asymmetriska nycklar i den inledande autentiseringsprocessen, mellan användaren och KDC:n, i exempelvis Kerberos.

(c) Hur löses problemet att en aktör skall kunna agera under olika identiteter? Identitets- och rollhantering stöds inte, då detta glider över på auktorisering, något som inte hanteras av Kerberos.

(d) Hur hanteras födelseprocessen? Oftast hanterar en administratör initieringen i ett system. Dock finns det inget som hindrar att detta skulle

kunna ske automatiskt och distribuerat, men till priset av minskad tillförlitlighet.

(e) Hur hanteras dödsprocessen? I centraliserade system underlättas dödsprocessen genom att den aktuella statusen för en användare måste verifieras vid inloggning. Hur pass aktuell informationen över användarna är beror helt och hållet på administratören och/eller de rutiner som finns för att uppdatera systemet. Det finns ingen teknisk fördröjning mellan ett uppdaterat system och verkan av uppdateringen.

Utifrån de ansatser om autonomitet som beskrevs i kapitel 3 kan Kerberos, och därmed biljettbaserade system i allmänhet, beskrivas enligt;

I. *Globala tjänster*. I Kerberos är nyckelservern (KDC:n) och biljettservern (TGS:en) centrala tjänster. Det är nödvändigt att en aktör får kontakt med dessa för att kunna autentisera sig och få tillgång till eftersökt tjänst.

II. *Förutsedd autonomitet*. Indelningen i realm:s är en segmentering av aktörer och tjänster. Varje realm kan verka oberoende av andra realm:s, dock kan en aktör endast nå de tjänster som erbjuds inom aktuell realm om kontakten med övriga realm:s är avskuren.

III. *Oförutsedd med styrd autonomitet*. Upprättandet av nya realm:s är inget problem. Varje ny KDC som etableras måste dock utbyta nycklar med andra KDC:er för att möjliggöra access till tjänster i andra realm:er. Kerberos v5 löser detta hierarkiskt genom att tillse att varje förfäder-realm delar nycklar med sina barn-realm. På så sätt finns det en väg mellan alla realm:er inom hierarkin. Lösningen förutsätter att det i ett första skede finns en logisk väg mellan egen realm (KDC) och sökt realm (KDC). När väl kontakten är etablerad har de egna KDC:n möjlighet att spara den nyckel som ger direkt kontakt med eftersökt KDC och på så sätt skapa en genväg genom hierarkin [NEU94].

IV. *Oförutsedd, icke styrd autonomitet*. Kerberos är väldigt känsligt för störningar i kommunikationen. En avgränsad realm kan mycket väl fungera även om angränsande realm:er är satta ur funktion, dock finns det bara möjlighet att nå de tjänster som finns inom egna realm:en. Sett ur ett av de tänkta scenariona i NBF där en aktör utan förberedelse skall kunna etablera kontakt med en ny, ej tidigare utnyttjad, tjänst är detta inte möjligt. Kerberos måste kunna kommunicera hierarkiskt för att kunna utföra spontana handlingar.

Summerar man egenskaperna i biljettbaserade system märker man ett stark beroende av globala tjänster, något som inte är önskvärt i exempelvis ett nätverksbaserat försvar. Inga av de nämnda systemen besitter någon högre grad av flexibilitet vilket gör att de inte är lämpliga att använda i system där förutsättningarna snabbt och kanske oförutsett ändras.

5.2 Certifikatbaserad autentisering: Public Key Infrastructure

5.2.1 Allmänt

Principen för autentisering med hjälp av publika nycklar utgår ifrån asymmetriska krypteringsmetoder. Med asymmetri menas att man använder två olika nycklar för signering och verifiering. Signeringsnyckeln är en hemlig nyckel i det avseende att endast en aktör skall ha tillgång och kunna utnyttja denna. Verifieringsnyckeln är publik, vilket innebär att den kan spridas fritt utan att äventyra säkerheten i systemet. Säkerheten i systemet bygger på att det skall vara praktiskt omöjligt att med hjälp av verifieringsnyckeln kunna återskapa signeringsnyckeln och därmed kunna maskera sig som en autentisk avsändare.

Ovanstående resonemang ställer automatiskt stora krav på hur signeringsnyckeln förvaras och utnyttjas. Ett vårdslöst hanterande av signeringsnyckeln kan medföra att den avslöjas eller på annat sätt exponeras vilket gör maskering som autentisk avsändare fullt möjligt.

Även om verifieringsnyckeln är ”säker” att sprida är det ändå svårt för en mottagare att säkert veta vem nyckeln tillhör. Loren Kohnfelder [KOH78] introducerade 1978 en lösning på detta problem. Genom att utnyttja digitala signaturer kan man associera ägarinformation till verifieringsnyckeln och på så sätt vara säker på vem eller vad som knyts till aktuell verifieringsnyckel. Allmänt kallas detta för digitala certifikat.

Exakt vilken information som binds till verifieringsnyckeln varierar. I det enklaste fallet associeras verifieringsnyckeln med en elektronisk mailadress. Detta är fallet i PGP (se exempelvis [PGP]). En mer uttömmande associering görs i X.509 (se [HOU02] standarden, där mer information om ägaren inkluderas i certifikatet vid registreringen.

Ett certifikat utfärdas för en viss tid. Hur lång denna tidsperiod är varierar beroende på nyttjandeområde men praxis inom kommersiella system, t.ex. e-bank, är ett år. Nu kan det mycket väl hända att ett certifikat måste förklaras ogiltigt inom sin levnadstid. Orsakerna till en ogiltigförklaring kan vara många, t.ex. att signeringsnyckeln exponeras.

Asymmetriska nycklar och certifikat är grundläggande enheter i Public Key Infrastructure (PKI). PKI är ett övergripande namn över hur man kan hantera asymmetriska nycklar i ett system för att kunna uppnå en säker autentisering och kommunikation. I [BEN01] presenterades några olika PKI lösningar belysta i nätverksbaserade system. En slutsats från [BEN01] var att X.509, i sin ursprungliga variant, bedömdes som stelbent att implementera och förvalta. Dock kommer X.509 att användas i denna rapport som en beskrivningsmodell beroende på att strukturen är tydlig och att flera egenskaper från X.509 känns igen i andra PKI lösningar.

De olika PKI-lösningarna diskuteras också av [GOL99], som observerar att det finns ett spektrum av PKI-lösningar. I den ena änden av spektrum befinner sig X.509 med en global certifikatbas, som är tung att förvalta. I den andra änden av spektrum finns PGP som baserar sig på användares/aktörers rekommendationer av varandra, strängt taget utan någon certifieringsmyndighet. Denna certifikatkedja, baserad på rekommendationer, kan med andra ord innebära att aktör A kan

komma att ge access till aktör C, på grund av B:s rekommendation av C, trots att A de facto inte önskar interagera med aktör C.

Mellan dessa ytterpunkter befinner sig SPKI/SDSI, vilken är en sammanslagning av två separata projekt SPKI (Simple Public Key Infrastructure) [SPKI] och SDSI (Simple Distributed Security Infrastructure) [SDSI]. [BEN01] lyfte fram denna PKI-lösning som speciellt intressant.

X.509 som beskrivningsmodell

Den centrala enheten i ett X.509 system är *Certifieringsenheten* (Certification Authority, CA), vilkens huvuduppgift är att signera certifikat samt att lista upphävda certifikat. En CA signerar certifikat med sin signeringsnyckel. Det medför att CA:ns, och därmed hela systemets, signeringsnyckel är mycket känslig för exponering. Detsamma gäller för alla signeringsnycklar i asymmetriska system, men jämfört med en signeringsnyckel hos en enskild användare påverkar CA:ns nyckel flera andra instanser om den skulle avslöjas.

Säkerheten för CA funktionen är alltså kritisk, vilket har medfört att man gärna flyttar ut mindre kritisk funktionalitet utanför CA:ns område. Exempel på detta är registrering och lagring av certifikat. Dessa funktioner är mindre säkerhetskritiska i olika avseenden. Just separation av tjänster är grundläggande egenskaper i system som gör anspråk på att vara säkra. Genom att skilja signeringen från, i sammanhanget, triviala procedurer och tjänster uppnår man en högre säkerhet för CA:ns privata nyckel.

I kapitel 4.1 beskrevs problematiken med att introduceras i ett system. Introduktionen i ett PKI system kan hanteras av *Registraturen* (Registration Authority). En användare eller enhet som behöver ett certifikat måste först genomgå en identifieringsprocess i syfte att säkerställa en korrekt identitet för certifikatet. Bevisföringen varierar, som nämndes ovan, beroende på syftet med certifikatet. Först när registreringen är klar signerar CA:n certifikatet.

Det signerade certifikatet lagras ofta i en *lagringsenhet* (Repository). Kraven på säkerhet i lagringsenheten behöver inte vara så höga, då integriteten i ett certifikat garanteras av CA:ns signatur.

Online - offline

Mycket av den verksamhet som sker över Internet är serverbaserad. Det innebär att en klient måste ha en etablerad kontakt, vara online, med en server innan en tjänst kan utföras. I ett autentiseringssystem kan autentiseringstjänsten utföras online eller offline, beroende på vilken typ av mekanism som används.

Ett online system har den fördelen att informationen kan vara mer aktuell, dock är det ingen garanti för detta. Nackdelen är dock att tjänsten blir beroende av en eller flera servrar, vilket kan medföra att en klient nekas en tjänst om kontakten är dålig eller bruten.

Offline system å andra sidan är mindre känsligt för störningar i kommunikationen. Kostnaden för denna tålighet är risken för mindre aktuell information. När det gäller autentisering kan det vara så att ett certifikat har återtagits sedan klienten senast uppdaterade sin återtagningsinformation.

I denna rapport kommer de olika återtagningsmetoderna att grupperas efter dess beroende av en yttre server, dvs. om de arbetar online eller offline.

Systemekonomi

Ett annat sätt att jämföra metoder är att studera hur pass kostsamma de är för systemet. Kostnad avser i de här fallen det lokala beräkningsbehovet samt den mängd information som behöver transporteras mellan olika enheter.

Kryptering är beräkningskrävande, i synnerhet asymmetrisk kryptering. Att verifiera ett certifikat är visserligen mindre krävande än vanligt kryptering, med ställer ändå krav på den enhet som skall utföra verifieringen. Primärt kanske detta inte är ett problem för en enskild aktör, men om aktören är en tjänsteproducent kommer flera verifieringar att göras och därmed kommer också beräkningsbelastningen att märkas.

System där mycket information måste transporteras mellan olika enheter kan snabbt bli överbelastat. Risken är stor att information försvinner längs vägen vilket kan vara förödande om det är kritisk information.

5.2.2 SPKI/SDSI

I [BEN01] framträder SPKI/SDSI som en intressant PKI-lösning relativt de olika delkraven A-F. Detta gäller bland annat kravet om möjligheter för avknoppning av autonoma delsystem (delkrav E). Gruppbegreppet, vilket är en grundläggande del av SPKI/SDSI, utgör en tydlig fördel för att realisera avknoppning av autonoma delsystem.

SPKI/SDSI baseras, i motsats till X.509:s globala bas, på en decentraliserad certifikatbas. Till varje publik nyckel kan en lokal certifikatbas, som är relaterad till denna nyckel, genereras. Respektive certifikatbas utgör en grupp. Ett antal sådana certifikatbaser kan sammanlänkas till en, enligt [CLA99], flexibel och kraftfull PKI-lösning, en struktur av samverkande grupper som kan underlätta realiseringen av ett nätverksbaserad försvar.

Namn- och auktorisationscertifikat

SPKI/SDSI tillhandahåller två typer av certifikat, namncertifikat och auktorisationscertifikat. Namngivning och auktorisation är separerade, och därmed undviks en uppsättning problem man annars får, enligt [ELI98], mellan namngivning (bindningen av något subjekt till en identifierare) och auktorisering (att delegera rättigheter till något subjekt). Autentisering och namngivning är nära relaterade problemområden.

Denna rapports fokus på autentisering, medför större viktläggning av frågor kring namncertifikaten än frågor kring auktorisationscertifikaten. Vissa av de formulerade problemställningarna och kraven som diskuteras kan dock delvis hitta sin lösning med hjälp av auktorisationscertifikaten. Av denna anledning presenteras och diskuteras båda certifikattyperna.

Ett namncertifikat C kan beskrivas som C=(K, A, S, V) där:

- K är en publik nyckel knuten till den digitala aktör som signerar certifikatet

- A är en identifierare av aktören, t.ex. *A, B, Alice* eller *Bob*.
- Subjektet S specificerar aktören närmare, t.ex. Alice Ted, vilket är en angivelse av ”Ted med relation till Alice” eller motsvarande. Aktörens verifieringsnyckel kan specifieras som del av S.
- V är en validitets-/giltighetsspecifikation. Anges normalt som ett tidsintervall.

En viktig egenskap med SPKI/SDSI är att den med sin lokala certifikatbas också baserar sig på lokalt unika namn. Vad som i PKI-litteraturen diskuteras som ”John Smith-problemet” kan därmed minskas eller undvikas. Om ett antal likartade namn (t.ex. John Smith) finns i en stor bas, som X.509, ger detta en risk för förväxlingar. Vikten av lokalt unika namn är en av relativt få PKI-frågor där samstämmighet ser ut att råda mellan experter på området, vilket framgår av [CHA01]. [CHA01] redovisar arbetet med ett expertsystem för att beräkna graden av förtroende till bindningen namn-publik nyckel. Nämnda studie baserar sig på frågeformulär och intervjuer utgående ifrån dessa med ett antal välrenommerade PKI-experter.

Hanteringen av lokala namn i SPKI/SDSI diskuteras med ett tekniskt fokus i [CLA99], och med ett delvis mera tillämpningsnära perspektiv i [DOH02]. Den typ av situation som [DOH02] speciellt diskuterar är mindre och samarbetande grupper, samt vikten av att namn bör vara knutna till en situation och konstellation (exempelvis arbetsgrupp) där namngivning inte introducerar en förväxlingsrisk. Om en förväxlingsrisk uppstår är det dock smidigare att hantera detta i en liten grupp än globalt.

Detta fokus på lokala arbetsgrupper relaterar tydligt till NBF-scenariot med mindre och ibland autonoma grupper, även om just NBF inte direkt diskuteras i artikeln.

Autentiseringsproblematiken är mera avgränsad än auktoriseringsproblematiken. Kort formulerad är det en fråga om man innehar korrekt identitet eller inte. Det finns dock en del intressanta varianter av autentisering som har studerats mindre. I synnerligen dynamiska situationer som kan uppstå i NBF-scenariot kan det bli aktuellt att olika aktörer behöver autenticeras till olika tider, men i samma ärende. Man kan tänka sig detta realiseras i form av en *stafettpinne* som går från aktör till aktör. Behov finns här av att autentisera stafettpinnen och aktör som innehar den, men också klargöra koppling mellan dessa. I utgångspunkten bedömer vi att gruppbegreppet och andra mekanismer i SPKI/SDSI bör underlätta även sådan autentisering.

Stafettpinnescenariot är en inriktning som bör studeras närmare.

Ett auktorisationscertifikat kan beskrivas som C=(K, S, d, T, V) där:

- K är en publik nyckel knuten till den digital aktör som signerar certifikatet. Denna aktör som utger certifikatet beviljar en specifierad auktorisation.
- Subjektet S. De publika nycklerna som ingår i värdemängden $\nu(S)$ är de som mottar auktorisation.
- Delegationsbiten d, om d= Sann, ger varje nyckel i $\nu(S)$ vidare delegeringsrättighet till andra.

- T är en auktoriseringsspecifiering, till exempel att man ges filaccess till en viss fil, inloggningsmöjlighet till specifierad dator eller annat.
- V är en validitets-/giltighetsspecifikation. Anges normalt som ett tidsintervall.

Auktorisationscertifikaten specificerar aktörers rättigheter och delegeringar. Vi bedömer att detta rent teknisk ytterligare torde underlätta avknoppningsproblematiken, ty vid olika typer av avknoppning är det av vikt att klargöra tydligt vilka rättigheter som är delegerade. Likaså är det av vikt att klargöra vilka rättigheter som inte är delegerade. Ovan beskrivna stafettpinnescenario torde också lättare realiseras med hjälp av SPKI/SDSI:s auktorisationscertifikat.

Autonomitetsegenskaper

(a) Autentiseringsprotokollet: Äktheten i digitala certifikat baseras på förmågan att verifiera digitala signaturer. Signeringen bekräftar att verifieringssnyckeln och tilläggsinformationen är knutna till varandra.

(b) Hopkoppling identitet-verifieringsnyckel: Olika varianter av detta finns för olika PKI-varianter. SPKI/SDSI-varianten med namncertifikat, som knyts till en publik nyckel med eventuell tilläggsinformation som ytterligare klargör identiteten, bedömer vi har flera fördelar.

(c) Aktör under olika identiteter: Om man med identitet här avser roller, kan man observera att detta i SPKI/SDSI kan hanteras med hjälp av gruppbegreppet och auktorisationscertifikat.

(d) Födelseprocessen: SPKI/SDSI låter initiering av nya användare delegeras genom att enskilda publika nycklar kan agera som CA. Detta ger en gruppstruktur som vi bedömer underlättar för realisering av ett nätverksbaserad försvar.

(e) Dödsprocess och spärrlistor: En omfattande mängd av lösningar finns. Problemet diskuteras separat. Standardlösningen i SPKI/SDSI är en validitets- och giltighetsspecifikation, men inte separata spärrlistor. Spärrlistor kan dock också hanteras inom SPKI/SDSI.

De olika aspekterna av autonomitet kan kommenteras enligt följande:

I. *Globala tjänster*: Existerar behov av att bygga striktare hierarkier med globala tjänster, så kan detta byggas genom en sammanlänkning av lokala certifikatbaser till en större struktur. Grundidéen i SPKI/SDSI är dock en mera distribuerad struktur med lokal bestämmanderätt. Detta gör att direkta beroenden av globala tjänster kan undvikas, vilket är en fördel i ett nätverksbaserat försvar.

II. *Förutsedd autonomitet*: SPKI/SDSI är uppbyggd kring grupper, i och med att varje publik nyckel kan agera som CA och att ett antal sådana CA-enheter kan byggas ihop till en större struktur. Omstrukturering kan också genomföras smidigt. Detta möjliggör den segmentering som behövs inför en förutsedd autonom situation med avseende på bland annat autentisering. Tillsammans med möjligheter för att även effektivt hantera delegering och auktorisering är detta faktorer som talar för SPKI/SDSI

III. *Oförutsedd, men styrd autonomitet*: I de fall man har hunnit klargöra delegationer innan avbrottet, till exempel med hjälp av auktorisationscertifikaten i SPKI/SDSI, kan avknoppningen hanteras. Däremot uppträder problem om delegation inte har definierats innan avbrottet. Den sista situationen väcker frågor om hur delegering och avknoppning totalt bör hanteras. I kritiska situationer har man inte tid och möjlighet att klargöra förhållanden kring sådant som inte har utretts tidigare.

IV. *Oförutsedd, icke styrd autonomitet*: SPKI/SDSI bedömer vi kan hantera denna situation om tillräckligt nya certifikat har givits och om de är giltiga under den tid man är autonom. Dock riskerar man att detta inte är fallet, och därmed betydande osäkerhet uppträder. Den lokalt baserade struktur SPKI/SDSI har kan dock underlätta, till exempel om den eller de närmaste CA-enheter man behöver kommunicera med fortfarande är tillgängliga.

5.2.3 Certifikatåtertagning

När ett certifikat utfärdas anges ett slutdatum då certifikatet upphör att gälla. Giltighetstiden varierar t.ex. beroende på utfärdaren eller syftet med certifikatet. Dock finns det tillfällen då det kan vara nödvändigt att dra in giltigheten för certifikatet i förtid. En sådan situation kan uppstå t.ex. om signeringsnyckel till ett certifikat blir stulen och därmed möjliggör för en annan person att maskera sig som den identitet certifikatet skall bekräfta.

Av denna anledning finns det tekniker för att sprida information avseende ogiltiga certifikat. Den ursprungliga lösningen, i X.509, var att publicera en lista över ogiltiga certifikat. Denna lösning används ännu, men varianter och alternativ har framtagits då dessa listor har flera svagheter. Kritik har riktats mot återtagningsmodellerna, då de endast ger negativa svar. En enhet kan endast bekräfta att ett certifikat har blivit återtaget – jämfört med en bekräftelse av att det fortfarande gäller.

I det här kapitlet presenteras certifikatåtertagningslistor och dess varianter, samt alternativ till dessa. De skiljer sig i uppdateringsintervall, uppkopplingsberoende och skalbarhet, men även strukturmässigt.

Certificate Revocation Lists

Den vanligaste och även den mest kritiserade modellen för att upphäva ett certifikats giltighet är att publicera listor över indragna certifikat. Dessa listor sammanfattas under namnet Certificate Revocation Lists, CRL. Listorna publiceras av CA, eller av någon delegerad enhet, och lagras tillsammans med certifikaten i lagringsenheten.

En CRL anger normalt:

- Listans serienummer
- Listans producent (issuer name)
- Tid för producerande (issuer time)
- Nästa beräknade uppdatering
- Indragna certifikat med

- Serienummer
- Tidpunkt för indragande

Det är uppenbart att en sådan här lista kan växa till ett mycket stort dokument i ett stort system. Certifikat som har gått över sin giltighetstid tas inte upp i CRL:en då de automatiskt avfärdas av den enhet som tar emot certifikatet.

En CRL anger egentligen den senaste tidpunkten då man kunde lita på certifikatet. Tiden från producerande tills nästa uppdatering är en gråzon. En mottagare kan inte vara säker på att ett aktuellt certifikat är giltigt bara för att det inte var angivet på senaste CRL:en. Det är alltså tveksamt om man kan avgöra den aktuella statusen på ett certifikat. Vidare så kan man inte undersöka statusen på ett enskilt certifikat, utan vid förfrågan skickas hela listan, vilket är kostsamt med avseende på bandbredd.

Ett system belastas hårt vid det tillfälle en ny CRL distribueras. Både storleken och intervallet avgör hur stor belastning det blir.

För att avhjälpa denna ovisshet har ett antal varianter av CRL framkommit där listan delas upp med avseende på volym och tid.

Delta-CRL

En delta-CRL [HOU02] är en mindre lista med de senaste indragningarna. Den ersätter inte den vanliga CRL:en, utan kompletterar den. Under tiden mellan uppdateringar av den vanliga CRL:en kan det komma flera delta-CRL'er. Varje delta-CRL är knuten till en given CRL, dvs. den skall kunna ange aktuell CRL:s serienummer.

För att ha kontroll över indragna certifikat måste man ha tillgång till den senaste CRL:en och den senaste delta-CRL:en. Resultatet blir en mer aktuell information över indragna certifikat, men problemet med gråzonen kvarstår, om än i mindre skala.

Indirekt CRL

Indirekta CRL:er möjliggör kombinationer av flera producenters CRL:s. Detta kan vara lämpligt om man vill konstruera en kombinerad Authority Revocation List (ARL) vilken listar samtliga indragna CA certifikat inom en viss domän (se t.ex. [ANK00]).

CRL Distribution points

Belastningen av nätet ökar kraftigt vid varje nytt utfärdande av en CRL. Flera försök att reducera trafiken momentant, och därmed sprida den över tiden, har gjorts, däribland CRL Distribution points (CRL-DP) [ADM98]. I CRL-DP anges en maximal storlek av varje CRL vilket i praktiken medför en segmentering av listan. Dessa segment sprids sedan över flera noder och därigenom fördelas trafiken jämnare över nätet.

Over-issued CRL:s

Ett annat sätt att angripa problemet med höga toppar i distributionen är att använda överlappning. Genom att producera flera listor med olika levnadstider kan man fördela nivån på trafiken och därmed sänka belastningen över nätet.

[COO99] föreslår en modell med over-issued delta CRL:er, vilket borde ge en jämnare belastning över nätet men med bibehållen aktualitet.

Online Certification Status Protocol (OCSP)

En av de stora nackdelarna med listor är att det alltid finns ett större eller mindre spann mellan aktuell information och gråzonen innan en ny uppdatering görs. Vidare kan en aktör inte bekräfta giltigheten i ett certifikat, utan hela listan lämnas som svar. En metod för att förmedla aktuell och preciserad information är Online Certification Status Protocol (OCSP).

OCSP är ett realtidsprotokoll för servrar. Förfrågningar hanteras av OCSP Responder vilken kan besvara förfrågningar över specifika certifikat, dvs. en fråga – ett svar. I det ursprungliga protokollet [MYE99] är OCSP endast en buffert mellan en aktör och, exempelvis, en CRL. Detta medför att färskheten i ett svar från OCSP inte behöver vara bättre än i ett system med CRL:er.

Ett förslag har framkommit [MYE01] där OCSP Respondern skall placeras tillsammans med en CA. Tanken är att CAn kontrollerar OCSP Respondern och på så sätt möjliggör att all information över återtagna certifikat som finns hos CA:n förmedlas direkt via OCSP Respondern. Denna lösningen medför hög aktualitet i svaret.

Certificate Revocation Status Directory (CRS Directory)

[MIC96] har studerat återtagningsmodeller där CRL:er används och kommit till slutsatsen att det sker lite trafik mellan CA och lagringsenheten, medan trafiken är mycket intensivare mellan lagringsenheten och enskilda användare. Konsekvensen, när det gäller CRL:er, blir att nätet belastas hårt då mycket information skall spridas. En studie som Micali, och även andra CRL kritiker, hänvisar till är [BER94]. Enligt studien kommer cirka tio procent av alla certifikat att återtas. Detta innebär att en CRL efter hand kommer att växa till ett stort dokument.

Genom att istället utöka den information som skickas mellan CA och lagringsenheten lyckas Micali minska informationsmängden som skickas mellan lagringsenheten och en slutenhet vid certifikatverifiering. I praktiken innebär detta att två 100 bitars värden, ett Y-värde för YES och ett N-värde för NO, tillförs certifikatet vid genereringen. Dessa värden införs i samband med att CA:n skapar certifikaten, genom att CA:n slumpar fram två basvärden Y_0 och N_0. Därefter modifieras Y_0 genom att utföra en envägs hashfunktion, F, så många gånger som certifikatet är giltigt i dagar. Exempelvis $Y = F^{365}(Y_0)$ för ett certifikat giltigt i 365 dagar. N skapas också genom att använda en envägs hashfunktion, men endast en gång ($N=F(N_0)$) oavsett giltighetstiden.

Vid varje uppdateringstillfälle skickar CA:n ett 100 bits värde *x*, motsvarande dagens F-värde, för varje certifikat till lagringsenheten. Om certifikatet fortfarande är giltigt skickas $F^{365\text{-}idag}(Y_0)$. Om certifikatet är ogiltigt skickas N_0.

När en aktör vill verifiera ett certifikat skickar lagringsenheten värdet *x* till aktören. Aktören, i sin tur, utför en hashfunktion en gång på *x* för att avgöra om $F(x) = N$. Stämmer inte $F(x)$ och N, hashas *x* vidare så många dagar som certifikatet varit giltigt. Tillsist skall Y erhållas. Om inte så kan användaren misstänka att lagringsenheten eller CA:n är otillförlitliga.

I och med att det är endast CA:n som känner till värdena Y_0 och N_0, är det praktiskt omöjligt för lagringsenheten att förfalska *x*. Likaså avslöjas lagringsenheten, eller om någon maskerar sig som denne, i de fall då ett gammalt *x* skickas. Antalet gånger envägsfunktionerna behöver köras är direkt kopplat till antalet dagar som certifikatet har varit giltigt.

En av de stora fördelarna med CRS är att en användare får ett positivt svar, dvs. att man kan verifiera att ett certifikat är de facto giltigt, vid senaste uppdateringstillfället, jämfört med CRL där man endast vet om ett certifikat har återkallats. Vidare blir mängden information vid en förfrågan avsevärt mindre än med ett CRL system.

Hierarchical Certification Revocation Scheme (HCRS)

Hierarchical Certification Revocation Scheme (HCRS) [AIE98] är likt CRS i det avseende att mängden verifieringsinformation ingående i certifikatet har utökats. Dock så har mängden information minskats jämfört med Micalis modell, främst genom att den nödvändiga återtagningsinformationen är organiserad i en trädstruktur. Resultatet blir att storleken på informationen som skickas mellan CA:n och lagringsenheten växer logaritmiskt med mängden certifikat.

Varje löv, dvs. ett indraget certifikat, i trädet tilldelas en slumpmässig sträng (*r*) av längden *l*. Föräldern till lövet beräknas med en envägs hashfunktion, F, sådant att *l* minskar sin längd med 1 för varje övre nivå som beräknas (*r*, F(*r*), $F^2(r)$, …, $F^D(r)$). *D* är antalet giltiga dagar för certifikatet och således också höjden på trädet. Trädet är komplett när roten är tom. Samtliga värden i kedjan av *r* inkluderas i certifikatet.

CA:n skickar vid varje uppdatering en lista R_i med indragna certifikat. Listan innehåller *dag-i verifieringsnoder* dvs. de högsta (närmast roten) noder i trädet som är giltiga enligt följande två regler;

1. För varje cert v $\notin R_i$ finns det minst en nod längs vägen från v (lövet/certifikatet) till roten som är en *dag-i verifieringsnod.*
2. För varje certifikat w $\in R_i$ finns det ingen nod som är en *dag-i verifieringsnod.*

Antag att inga certifikat har dragits in första dagen. I detta fallet skickar CAn roten till lagringsenheten, vilket medför att alla certifikat kan verifieras som giltiga då roten ingår i alla certifikat. Om man istället antar att certifikat 010 har upphävts, så kommer samtliga förfäder till 010 (inklusive 010), dvs. {01, 0 samt roten}, att markeras som upphävda. En *dag-i* lista kan då innehålla verifieringsnoderna {00, 0100, 0101, 011 samt 1}. Att en förfäder till ett certifikat är upphävd betyder således inte att alla barn till denne är upphävda, då regel 1 säger att för varje giltigt certifikat finns det minst en *dag-i verifieringsnod*. Ett upphävt certifikat saknar verifieringsnoder, då det är den kortaste vägen mellan löv och rot som är markerad som upphävd.

Certificate Revocation Trees (CRT)

En av fördelarna med CRL:er, jämfört med on-line verifiering, är att en CRL endast behöver signeras en gång. Vid on-line verifiering signeras varje svar. Den huvudsakliga fördelen med on-line verifiering är att endast ett certifikat verifieras

vilket minskar trafiken mellan lagringsenheten och slutanvändaren. Dessa fördelar utnyttjas i Certificate Revocation Trees (CRT) (beskrivs bl.a. i [WIL99] och [BER98]).

Systemet bygger på idén att alla certifikat utfärdade av en given CA kan sorteras i nummerordning och grupperas i kluster av giltiga certifikat. För man in denna idé i en binär trädstruktur får man att alla löv är intervall med giltiga certifikat utom det första, vilket är upphävt. Genom att utföra en envägs hashfunktion på två löv (syskon) erhåller man föräldern till dessa. Fortsätter man med att hasha föräldern och dess syskon (lövens farbröder) får man lövens farförälder och så vidare. Slutligen når man en rot vilken är stamfader till hela trädet. Kocher visar att det är tillräkneligt att signera roten i trädet, då envägs hashfunktioner gör det praktiskt omöjligt att konstruera trädet enbart med hjälp av roten.

I praktiken skapar CA:n hela trädstrukturen och signerar roten. Därefter skickas den signerade roten tillsammans med en osignerad lista till lagringsenheten. Med hjälp av listan skapar lagringsenheten en egen kopia av trädet samt verifierar signaturen av roten. När en enhet vill verifiera ett certifikat skickas den signerade roten, aktuell nod (det intervall certifikatet tillhör), nodens syskon samt nödvändiga noder (förfäder) vilka behövs för att återskapa roten. Stämmer den återskapta roten med CA:n signerade rot är verifieraren säker på att informationen är korrekt. Befinner sig det sökta certifikatet inom intervallet på aktuellt löv, med undantag av lövets nedre gräns, är det giltigt, annars är det upphävt.

CRT är en mycket beräkningskrävande modell. Varje förändring av löven kräver att hela trädet räknas om.

Noar and Nissim’s scheme

Ett av syftena med att använda sig av lagringsenheter ect. är att minska interaktionen med CA:n och därmed minska risken för att CA:ns hemliga nyckel skall exponeras. Noar och Nissim [NOA98] tog fasta på detta när de publicerade sin lösning på certifikat upphävande. Släktskapen med CRT och CRL är tydliga då de bygger sin lösning på en trädstruktur där de interna noderna är hashsummor och löven är återtagna certifikat. Modellen verifierar således att ett certifikat har blivet återtaget och indirekt om ett certifikat fortfarande är giltigt.

Deras schema bygger på ett balanserad 2-3 trädstruktur, vilket innebär att varje intern nod har två eller tre barn och att sträckan från rot till ett löv är densamma för alla löv. Varje löv är ett återtaget certifikats serienummer. När CA:n uppdaterar trädet skickas en lista med de certifikat som har ändrad status (dvs. skall läggas till eller tas bort) samt en signerad rot och trädets höjd. Lagringsenheten justerar löven i sitt träd enligt CA:ns lista samt räknar om trädet. Att informationen är korrekt kan lagringsenheten verifiera med den signerade roten och trädhöjden. När en användare vill verifiera ett certifikat vidarebefordrar lagringsenheten den signerade roten och trädets höjd. Är certifikatet återtaget finns det en väg från det givna lövet till roten, vilket medför att användaren kan återskapa trädet och verifiera roten om lagringsenheten även skickar med värdet på alla syskon längs vägen. I de fall där certifikatet fortfarande är giltigt finns det inte med som något löv i trädet. En användare verifierar detta genom att få tillgång till vägen för närmaste löv med högre och lägre värde. På så sätt kan användaren verifiera att det finns ett lägre och ett högre serienummer som har återtagits, samt att det inte finns några återtagna certifikat däremellan.

Noar & Nissim's modell reducerar kommunikationskostnaden mellan CA och lagringsenheten drastiskt jämfört med CRT. Kommunikationen mellan lagringsenheten och användaren är ungefär den samma som med CRT.

Sammanfattning certifikatåtertagning

De modeller för certifikatsåtertagning som presenterats i denna rapport skiljer sig i uppbyggnad, uppdateringsintervall och kommunikationskostnad. CRL:er har hängt med sedan X.509 skapades och har sedan dess utsatts för hård kritik. Mycket av den kritiken har rört sig om ineffektiviteten i användandet av CRL:er, främst färskheten i informationen, samt det faktum att en CRL inte bekräftar giltigheten i ett certifikat. Den påvisar endast att ett certifikat inte har blivet indraget, dvs. ett negativt svar. Av de modeller som presenterats här är det endast CRS Directory som kan bekräfta att ett certifikat fortfarande är giltigt.

Termen lagringsenhet har används genom hela kapitlet. Dock är det något missvisande med avseende på online modeller. I mångt och mycket är lagringsenheten där en vanlig server, då den måste kunna utföra vissa beräkningar och fatta beslut om vilken information som skall vidarebefordras.

Av punkterna (a)-(e) i kapitel 4 är punkt (b) lika aktuell för lagringsenheten som för andra delar i ett PKI system. I återtagningssammanhang gäller detta främst för onlinemodeller, då de är en instans som kräver ett visst förtroende från användarna. I huvudsak är det dock punkt (e) *Hur hanteras dödsprocessen* som har beskrivits.

Rapporten har för avsikt att studera hur väl ett system kan hantera olika grader av autonomitet. I kapitel 3 beskrevs vissa grader av autonomitet som skulle beaktas;

I. *Globala tjänster*. Alla återtagningsmodeller är i viss mån beroende av globala tjänster. Lagringsenheterna är av CA:n delegerade enheter för praktisk förvaring och hantering av återtagningsinformation. CA:n är fortfarande den enhet som ansvarar för att en återtagning skall utföras, men lagringsenheterna möjliggör spridning av informationen.

II. *Förutsedd autonomitet*. Graden av beroende beror i hög grad av närheten till aktuell CA. Man kan vid design av systemet sprida CA:s och dess delegerade enheter strategiskt. Dock hjälper detta inte om återtagningsinformation behövs från en CA/lagringsenhet längre bort i systemet.

III. *Oförutsedd med styrd autonomitet*. Varken X.509 eller någon återtagningsmodell hindrar att man sätter upp nya enheter vid behov. Problemen man kan stöta på är de samma som i II, dvs. avståndet till informationen avgör tillförlitligheten i systemet.

IV. *Oförutsedd, icke styrd autonomitet*. Det segment som blir avskuret oförutsett har möjlighet att fungera obehindrat så länge en CA/lagringsenhet finns inom den kvarvarande domänen. Används CRL:er finns det även en möjlighet att fortsätta verifiera användare tillhörande den CA:n som utfärdat CRL:en. Dock kommer systemet att ”låsas” vid det informationsläge som var aktuellt vid tidpunkten för avskurningen. I praktiken betyder det att systemet måste acceptera vad det visste då och behandla alla nya förfrågningar som osäkra. Det ända som kan verifieras i detta läget är kända nycklar, exempelvis CA:ns signeringsnyckel.

Även andra ansatser har gjorts i syfte att undvika återtagningsmodeller som helhet alternativt minska behovet av dem. Ett sådant exempel är kortare giltighetstider på certifikaten. Detta skulle minska mängden återtagningsinformation över nätet, men samtidigt öka belastningen på CA:n då fler nya certifikat måste genereras.

Rivest föreslår i [RIV98] en ”självmordsbyrå” (SB) som ersättare till CRL:er. Idén är en vidareutveckling av PGP:s ”självmordsbrev”, ett brev som ägaren till ett nyckelpar skickar då signeringsnyckeln har blivit exponerad. Äktheten i brevet styrks av signeringen, även om signeringsnyckeln är exponerad. En användare registrerar sin verifieringsnyckel hos SB. Om signeringsnyckeln misstänks exponerad skickar ägaren ett självmordsbrev till SB, vilken sprider informationen vidare över nätet via broadcastmeddelande. För att förstärka tillförlitligheten i systemet kan ägaren till signeringsnyckeln skicka en ”hälsodeklaration” till SB:n i syfte att påvisa nyckelparets integritet. Denna hälsodeklaration kan SB vidarebefordra till andra instanser vid förfrågan om ett certifikats giltighet. En självmordsbyrå är tänkt att administrera ett begränsat nätverk, då SB:n avlyssnar nätverket efter självmordsbrev samt att broadcastmeddelanden i större utsträckning påverkar effektiviteten i nätverket negativt.

Ett antal argument mot användande av CRL-lösningar tas upp i [RIV98] där han redan i rubriken ställer frågan om vi kan eliminera CRL-lösningar. Huruvida denna kritik, och kritik från andra håll, allmänt håller analyseras i [MCD00]. Bland de problemområden och kritiska faktorer som tas upp kan nämnas:

- Vem bör ges rätt att ställa krav till hur nya certifikat måste vara, CA:n eller aktören som accepterar certifikatet?
- Var bör ansvaret för att tillhandahålla tillräckligt ny certifikatinformation ligga?
- Är certifikat med kort giltighetsperiod ett bra alternativ till CRL?

Enligt [MCD00] är svaret på Rivests grundfråga, om vi kan eliminera CRL-lösningar, både ja och nej, beroende på tillämpning och miljön kring tillämpningen. Exempelvis kan certifikat med kort giltighet vara ett alternativ till CRL:er, men detta ökar även trafiken i näten genom att nya certifikat skall förmedlas. Detta kan utgöra ett problem i exempelvis en NBF-situation med periodvis sämre nätkommunikation eller rent av avbrott.

Allmänt kan man observera, enligt [MCD00], att CRL-lösningar är svåra att hantera i stora, löst kopplade miljöer. I mindre lokala scenarion kan de i vissa fall vara fungerande. En lösning som [MCD00] som argumenterar för är möjligheten att beställa revokeringsinformation vid behov, ”revocation on demand”.

För NBF-miljön indikerar detta att ett helt försvarssystem bygd på revokeringslistor blir tungrodd, däremot kan det vara användbart i lokala lösningar, till exempel inom en mindre autonom grupp eller mellan ett fåtal autonoma grupper. Närmare fältstudier och simuleringar kan ge ytterligare information om när CRL-lösningar fungerar bäst eller när certifikat med kort giltighetsperiod är det bästa alternativet.

5.2.4 Kombinationsmetoder

I kapitlet om introducering och avveckling i ett system (kap 4.1) beskrivs i ett avsnitt "Administratörsberoende" fördelen, ur autonomitetssynpunkt, att inte vara beroende av en enda administratör. Där motiveras också att det inte är lämpligt att förbättra autonomiteten genom att inrätta två eller flera likvärda administratörer, så att en aktör vid behov kan kontakta en som råkar vara tillgänglig just då. Det man i stället vill ha, bygger på det som i kryptolitteraturen betecknas som "Secret-Sharing" [SCH96] eller "Threshold Cryptography" [KOT85]. Såvitt vi vet finns inga sådana metoder beskrivna som skulle kunna vara användbara i server-baserade respektive identitetsbaserade autentiseringssystem. Däremot finns ett antal metoder för digital signatur, vilket ju är grundvalen för PKI. Därför beskrivs, eller snarare skisseras, metoderna i detta kapitel.

Tanken är den, att det ibland, t ex i ett taktiskt läge med krav på korta tidsfördröjningar, kan vara svårt att komma i kontakt med en administratör, CA. Man behöver denna kontakt för att få nya certifikat, eller revokeringsinformation, för autentisering gentemot andra aktörer. Certifikaten, och revokeringsinformationen, måste vara digitalt signerad av betrodd instans för att man skall våga lita på informationen. Det är oftast lättare att få kontakt med andra, vanliga aktörer, i systemet än att få kontakt med CA. Då vore det attraktivt att kunna resonera ungefär så här: "Information signerad av en enda aktör vågar vi inte lita på, kanske inte heller av två, men om samma information signeras av k st särskilt betrodda aktörer är den godkänd".

Ett scenario, där kombinationsmetoder skulle kunna vara av värde, är att en mindre militär enhet, t.ex. en grupp, blir avskuren från sitt förband men kommer i kontakt med ett annat förband som gruppen inte tidigare haft kommunikation med. Aktörerna i gruppen behöver då snabbt certifikat som styrker identiteterna. Ett annat scenario är att delar av ett förband blir erövrat av motsidan. Då behöver revokeringsinformation om detta snabbt skapas, och man kanske inte hinner vänta på kontakt med CA.

När CA normalt sett skall signera ett certifikat, eller annan information, gör han detta genom att använda sin hemliga nyckel, h_{CA}, i en signeringsoperation på certifikatet. Alltså $\mathtt{sign}_{CA}=\mathtt{signering}(\mathtt{cert},h_{CA})$. Alla aktörer som känner till CAs publika verifieringsnyckel, pCA, kan sedan verifiera att certifikatet är korrekt signerat genom att använda den publika verifieringsnyckeln i en verifieringsoperation, $\mathtt{verifiering}(\mathtt{cert},p_{CA})=\mathtt{OK}$, där OK är ett kriterium som bevisar korrekthet.

Ett sätt att åstadkomma k st särskilt betrodda signaturer är naturligtvis att k st aktörer "på vanligt sätt" signerar informationen med hjälp av sina ordinarie signeringsnycklar. Men då måste den aktör, som vill verifiera informationen, själv avgöra om de k st signerande aktörerna är "särskilt betrodda just nu". Information om vilka aktörer som är betrodda måste alltså spridas till alla i systemet och hållas aktuell. Ett annat sätt, som beskrivs nedan, innebär att en CA styr vilka aktörer som anses betrodda. De verifierande aktörerna behöver inte hålla reda på detta, utan de verifierar med hjälp av en enda, allmänt känd, verifieringsnyckel.

Kombinationsmetoderna innebär att CA väljer ut ett antal ombud - säg $O_1,\ldots,O_n$ – bland aktörerna. Han tilldelar sedan varje ombud var sin hemlig delnyckel – $h_1,\ldots,h_n$. När CA skapar dessa delnycklar utgår han från en hemlig totalnyckel

som enbart CA känner till, jfr h_{CA} ovan, där motsvarande verifieringsnyckel, p_{CA}, har gjorts känd av alla. När CA skapar dessa delnycklar, väljer han en algoritm där ett tal k, $1<k<n$, ingår som parameter. Effekten blir att om man har minst k st delsignaturer, från vilka k st ombud som helst, kan man kombinera ihop delsignaturerna till en totalsignatur vars korrekthet kan verifieras. Däremot räcker det inte med $k-1$ st eller färre.

På detta sätt kan alltså en CA välja ett antal ombud bland de aktörer, som han av någon anledning bedömer som lämpliga ombud. Om minst k st ombud signerar samma information bedöms den som pålitlig, annars inte. Det finns ett antal variationer på detta tema. CA kan t ex välja en grupp ombud för en uppgift, en annan grupp ombud för en annan uppgift etc. CA kan också gradera pålitligheten bland ombuden genom att tilldela de mest pålitliga ombuden flera delnycklar. Ett sådant extra pålitligt ombud kan då bilda flera delsignaturer, och får därmed större vikt när k st delsignaturer skall kombineras.

Det bör påpekas att CA skall välja annan totalnyckel än den nyckel han använder för att ensam signera certifikat. För om k st ombud erövras av motsidan blir totalnyckeln, h_{CA}, röjd genom att kombinera de k st delnycklarna. Det räcker emellertid inte att motsidan får tag i k st delsignaturer. Det går nämligen inte att härleda delnycklar ur delsignaturer.

Det finns kombinationsmetoder publicerade för signering med RSA [GEN96a] och för DSS [GEN96b], Digital Signature Standard. I [FOK02] och [ZHO99] beskrivs fler aspekter på secret-sharing i ad-hocnät.

Slutsatsen av detta kapitel, är att kombinerade metoder är en tänkbar möjlighet att uppnå egenskaper som förbättrar autonomiteten.

5.2.5 Autonomitetsegenskaper

PKI som begrepp är ett samlingsnamn för skilda lösningar med vissa gemensamma faktorer. Centralt inom PKI är utnyttjandet av digitala certifikat, men metoderna skiljer sig strukturmässigt främst med avseende på hur mycket toppstyrning och beroende av andra enheter som krävs.

Kapitel 4 listade ett antal egenskaper avseende autonomitet. PKI har enligt den listningen följande egenskaper;

(a) Autentiseringsprotokollet: Äktheten i digitala certifikat baseras på förmågan att verifiera digitala signaturer. Signeringen bekräftar att verifieringssnyckeln och tilläggsinformationen är knutna till varandra.

(b) Hopkoppling identitet-verifieringsnyckel: Nyckel och identitet är bundna via CA:ns signering av certifikatet.

(c) Aktör under olika identiteter: Om man med identitet här avser roller, kan man observera att detta i SPKI/SDSI kan hanteras med hjälp av gruppbegreppet och auktorisationscertifikat. X.509 och PGP stödjer inte aktivt rollhantering, det finns inga inbyggda funktioner för detta i dessa system. Det finns dock inga tekniska hinder för att införa rollhantering i X.509 och PGP.

(d) Födelseprocessen: Nyckel och identitet måste kopplas till varandra och bekräftas av en betrodd part. Förtroende kan skapas centralt, som i X.509, eller byggas upp av relationer mellan aktörer, som i PGP och SPKI/SDSI. Kombinationsmetoder kan användas för att minska beroendet av en betrodd tredje part.

(e) Dödsprocess och spärrlistor: Ett certifikat skapas med en giltighetstid. Inom giltighetstiden kan inte en aktör tas bort, däremot kan vid behov aktörens accessmöjligheter spärras. För detta ändamål upprättas spärrlistor eller motsvarande.

De olika aspekterna av autonomitet kan kommenteras enligt följande:

I. *Globala tjänster*: Här framträder mycket tydligt spektrumet från globala, hierarkiska lösningar som X.509 till löst uppbyggda, decentraliserade lösningar som PGP. Det bör observeras att någon renodlad X.509-implementation med en enda CA högst i hierarkin inte existerar. Vad gäller beroende är det av global karaktär i X.509, vilket skapar en typ av sårbarhet som bör minimeras. I SPKI/SDSI framträder mera lokala beroenden genom att certifikaten är knutna till en lokal CA. I PGP existerar inga organisationsmässiga beroenden. Detta medför en risk att inte ges access, ty certifikatkedjorna garanteras inte innehålla önskad information.

II. *Förutsedd autonomitet*: Segmentering kan åstadkomma fungerande autonoma enheter i de olika diskuterade PKI-lösningarna. Deras möjligheter kan vara begränsade, men de kan fortfarande verka inom ramen av tillgängliga resurser. Genom att PGP saknar någon form av globalt beroende påverkar inte segmentering PGP:s förmåga.

III. *Oförutsedd, men styrd autonomitet*: SPKI/SDSI kan med sin lokalt baserade och icke-hierarkiska arkitektur anpassas till oförutsedda segmenteringsbehov. I och med att man inte är beroende av globala tjänster, är ett sådant system mindre känsligt vid autonomitet.

    I X.509 har aktörer en stark koppling till sin CA, vilken genererar revokeringsinformation. Vid styrd autonomitet kan det vara en fördel att skapa en ny CA för att ha kvar närhet till bland annat revokeringsinformation. Sökta tjänster kan dock fortfarande finnas långt borta i CA-strukturen, och därmed utom räckhåll vid autonomitet.

IV. *Oförutsedd, icke styrd autonomitet*: SPKI/SDSI och PGP har här fördelar i och med sina lokalt baserade arkitekturer. X.509 kan också fungera i dessa autonoma situationer, men möjligheterna att få korrekt information minskar. Kombinationsmetoder kan underlätta uppbyggnaden av oförutsedda grupper av aktörer.

Sammanfattningsvis observerar man i takt med ökande grad av autonomitet att X.509 med sin hierarkiska struktur får problem. Däremot fungerar SPKI/SDSI och PGP bättre med ökande grad av autonomitet. PGP saknar dock för mycket av den struktur som behövs inom en organisation. SPKI/SDSI kan fortfarande fungera tillräckligt väl.

5.3 Identitetsbaserad autentisering

Som nämndes i (kap 4) är en av de väsentligaste delarna i autentiseringsprocessen hur man knyter ihop identiteten med rätt verifieringsnyckel (fråga b)). En attraktiv tanke är frågan om inte verifieringsnyckeln kan vara lika med identiteten själv. I så fall behövs det ju inga register eller annat som kopplar ihop identitet och nyckel. Exempelvis vore det attraktivt att verifiera att ett e-mail kom från uppgiven avsändare genom att som verifieringsnyckel använda t.ex. avsändarens mailadress. Eller att en radionod, som begär access till ett radionät, kan verifieras med hjälp av radions identitet.

Nycklar är numeriska tal och för att autentiseringsmetoden skall få tillräcklig kryptologisk styrka finns det olika krav på nycklarna, de kanske t.ex. inte får innehålla gemensamma faktorer. Identiteter är å andra sidan ofta textsträngar, så därför går det sällan att få exakt likhet mellan verifieringsnyckeln och identiteten. Men det behövs inte, om det bara finns en av alla använd algoritm som en-entydigt översätter en identitet till en giltig nyckel, dvs. mot en identitet svarar en och endast en nyckel och omvänt. Sådana autentiseringsmetoder kallas identitetsbaserade metoder. I [BEN01] beskrevs två stycken, men det finns fler.

Vi vill påpeka att man kan tänka sig att koda in mycket information, förutom själva grundidentiteten (t.ex. ett namn), i det vi här kallar identitet. En identitet kan t.ex. i princip ha utseendet "Jag heter NN, lyder under administratör CA, min autenticeringsnyckel är giltig f o m datum1 t o m datum2, …". Detta under förutsättning att det finns en av alla känd matematisk funktion (jfr. hashfunktion) som kan översätta en sådan utökad identitet till ett numeriskt tal som uppfyller krav för att de kryptografiska algoritmerna skall bli säkra. En identitet skulle då i princip kunna jämföras med ett certifikat.

Verifieringsnyckeln är alltså den av alla kända identiteten. Det steg av övervakning som måste ske i födelseprocessen (fråga (d), kap 4) kan således inte knytas till verifieringsnyckeln, utan måste knytas till autentiseringsnyckeln. Denna kan därför inte aktörerna själva skapa, utan den måste skapas av, eller med hjälp av, en betrodd administratör. Det påpekades i [BEN01] att en väsentlig nackdel med de beskrivna metoderna är att alla aktörer måste administreras av samma administratör. Därmed blir det svårt att koppla ihop delsystem och det går alltså inte lätt att skala upp till stora system. Det får också konsekvenser för autonomiteten, i och med att administrationen inte kan delegeras.

I [BEN01] beskrevs två identitetsbaserade metoder. Den ena metoden, Guillou-Quisquater, baserar sig på publika nycklar, medan den andra metoden, Leighton-Micali, baserar sig på symmetriska hemliga nycklar. Vad gäller autonomitet bedöms de ha ungefär samma egenskaper. Därför görs ingen uppdelning dem emellan i denna rapport. De viktiga egenskaperna är att verifieringsnyckeln härleds ur identitetn själv, samt att administrationen måste skötas av en enda CA. För beskrivningar av metoderna hänvisas till [BEN01] och grundreferenserna [GUI89] resp [LEI94].

5.3.1 Autonomitetsegenskaper

De två identitetsbaserade metoderna har något olika egenskaper. Guillou-Quisqarter är relativt beräkningskrävande, medan Leighton-Micali kräver hantering av mer data, vilka dock inte behöver hållas hemliga. Detta innebär att de kan

ha olika för- och nackdelar i olika tillämpningar. Men i relation till frågeställningarna i kap y är de ganska likvärdiga. Den allt överskuggande egenskapen är att det i båda metoderna krävs en enda betrodd administratör. Därför har båda metoderna samma relationer till frågeställningara (a)-(e) om autonomitet.

(a) Autentiseringsprotokollet. Härvidlag finns inga principiella skillnader mot t.ex. PKI-baserade metoder.

(b) Hopkoppling identitet-verifieringsnyckel. Detta är metodens styrka, i och med att inga nyckelcertifikat eller annan liknande information behöver hanteras.

(c) En aktör under olika identiteter, i olika roller. Detta kan inte hanteras.

(d) Födelseprocessen. Denna kan inte delegeras, utan all administration måste skötas av en CA. Detta innebär att det bara kan finnas en uppsättning "lokala nät", vilket ger en inflexibel struktur. Utökning av ett lokalt nät med nytillkommande aktör sker genom att aktören kontaktar CA och tilldelas rätt autentiseringsnyckel.

(e) Dödsprocess och spärrlistor. Finns inte inbyggt i metoden. De kan hanteras via listor, i princip likvärdigt med de andra metoderna. Giltighetstider kan hanteras via kodning (se ovan), men det blir mindre flexibelt än t.ex. PKI-certifikat.

De olika aspekterna av autonomitet kan kommenteras enligt följande:

I. *Beroende av globala tjänster*. Det finns inget beroende av tjänster som är globala över systemet. Identitetsbaserad autentisering fungerar helt enkelt bara i lokala system.

II. *Förutsedd autonomitet*. Uppdelningen, vid design av systemet, av det totala systemet i olika segment är möjlig. Men den enda möjligheten till uppdelning är i form av lokala nät, som i princip är oberoende av varandra. Det är därför en ganska stel och oflexibel lösning i och med att andra alternativ saknas.

III. *Oförutsedd men styrd autonomitet*. En tillfällig enhet skall kunna sättas upp för att lösa en specifik uppgift, utan att detta varit förutsett vid designen av systemet. Denna tillfälliga enhet kan bestå av komponenter från flera olika delsystem, vilket medför att de måste ha möjlighet att skapa ett gemensamt förtroende. Den lösning som står till buds är att upprätta ett lokalt nät, administrerat av en enda CA.

IV. *Oförutsedd, icke styrd autonomitet*. Det kan kanske lösas på samma sätt som III ovan, dvs. ett lokalt nät med en CA. Administratören, CA, måste emellertid ha väldigt annorlunda kompetens jämfört med andra aktörer i det lokala nätet. Eftersom IV innebär att man inte har någon kontroll över vilka aktörer som skall ingå är risken stor att man inte kan hitta någon som kan fungera som CA.

6 Metodernas egenskaper vad gäller autonomitet

De tre klasserna av autentiseringsmetoder beskrevs i tre delavsnitt i kapitel 5. Vardera avsnittet avslutades med en diskussion om egenskaperna, inom respektive klass, vad gäller autonomitet. Nedan sammanfattas diskussionerna i den andra dimensionen - inom varje aspekt av autonomitet kommenteras de tre klasserna av autentiseringsmetoder.

I. *Beroende av globala tjänster*. Mest uttalat vid PKI enligt renodlad X.509-standard. Även serverbaserade (biljettbaserade) metoder kan ha beroende. För övriga metoder finns inget beroende av tjänster som är globala över systemet.

II. *Förutsedd autonomitet*. Uppdelningen, vid design av systemet, av det totala systemet i olika segment är möjlig i alla tre klasserna. Men serverbaserade metoder är, som alltid, beroende av en fungerande kommunikation. Den enda möjligheten till uppdelning vid identitetsbaserade metoder är i form av lokala nät, som ger en ganska stel och inflexibel lösning.

III. *Oförutsedd men styrd autonomitet*. En tillfällig enhet skall kunna sättas upp för att lösa en specifik uppgift, utan att detta varit förutsett vid designen av systemet. Denna tillfälliga enhet kan bestå av komponenter från flera olika delsystem, vilket medför att de måste ha möjlighet att skapa ett gemensamt förtroende. Detta kan åstadkommas i biljettbaserade system, men kräver då resurser i form av servrar och kommunikation. Den mest flexibla klassen av metoder är certifikatbaserad, under förutsättning att man kan konstruera ett tillräckligt bra system för spärrlistor. I identitetsbaserade system är den enda lösning som står till buds att upprätta ett lokalt nät, administrerat av en enda CA.

IV. *Oförutsedd, icke styrd autonomitet*. I lyckosamma fall, när t ex alla ingående aktörer råkar tillhöra samma CA, kan alla tre metoderna fungera. Men realistiskt sett är det bara certifikatbaserade metoder som kan rekommenderas. En speciell fördel är att man kan ta hjälp av kombinationsmetoder för att underlätta bildande av ad-hocnät.

6.1 Slutsatser

Bedömningarna av de tre autenticeringsklassernas egenskaper vad gäller olika aspekter av autonomitet kan mycket grovt sammanfattas i en tabell.

	Biljettbaserade	Certifikatbaserade	Identitetsbaserade
I. *Beroende av globala tjänster*	Beroende	Starkt beroende i renodlad X.509 Inget beroende i SPKI/SDSI o PGP	Inget beroende
II. *Förutsedd autonomitet*	Kan lösas via segmentering vid systemdesign	Löses väl via segmentering vid systemdesign	Kan lösas på inflexibelt sätt
III. *Oförutsedd men styrd autonomitet*	Kan gå bra under vissa förutsättningar	Löses inflexibelt i X.509 Löses flexibelt i SPKI/SDSI o PGP.	Kan gå bra under vissa förutsättningar
IV. *Oförutsedd ej styrd autonomitet*	Hög risk för misslyckande	PGP flexibelt men primitivt. Risk för misslyckande i de två andra, speciellt X.509. Kombinationsmetoder kan underlätta.	Kan gå bra, men hög risk för misslyckande

Tabell 6.1 Egenskaper vad gäller autonomitet

Bedömningarna ger en fingervisning om egenskaper inom den övergripande klassen, men variationer finns inom klassen. Exempelvis är certifikatklassen den mest anpassningsbara modellen. Man skall då ha i åtanke, att det finns flera olika implementeringar att välja på, och de ger olika resultat. Utifrån vårt material bedömer vi dock att certifikatmetoder (PKI) är de kandidater som är mest intressanta för de krav på autonomitet som finns inom det nätverksbaserade försvaret. Vi anser att SPKI/SDSI är den mest flexibla metoden. En stor nackdel är att den inte är implementerad och testad i samma grad som X.509 och PGP.

7 Referenser

[AIE98] Aiello W, Lodha S & Ostrovsky R; Fast Digital Identity Revocation, 1998. http://www.argreenhouse.com/papers/rafail/40.pdf, besökt 31 okt 2002.

[ADM98] Adams C & Zuccherato; A General, Flexible Approach to Certificate Revocation, 1998. http://www.entrust.com/resources/pdf/certrev.pdf, besökt 31 okt 2002.

[ANK00] Ankney R; Certification Revocation Mechanisms, 2000. http://cnscenter.future.co.kr/resource/rsc-center/vendor-wp/certco/revoc.pdf, besökt 31 okt 2002.

[BEN01] Bengtsson A, Hunstad A & Westerdahl L; Autentisering i nätverksbaserade system, FOI, 2001.

[BER94] Berkovits S, Chokhani S, Furlong J A, Geiter J A & Guild J C; Public Key Infrastructure Study – Final Report, 1994. http://csrc.nist.gov/pki/documents/mitre.ps, besökt 31 okt 2002.

[BER98] Berkovits S & Herzog J C; A Comparison of Certificate Validation Methods for Use in a Web Environment, 1998. http://www.mitre.org/support/papers/tech_papers_01/berkovits_comparison/berkovits_comparison.pdf, besökt 31 okt 2002.

[BIH92] E. Biham, ”New Types of Cryptanalytic Attacks Using Relating Keys”,Technical Report #753, Computer Science Department, Technion-Isreal Institute of Technology, September 1992.

[CHA01] Chadwick D W, Basden A, Evaluating trust in a public key certification authority, Computers and security vol. 20, no.7, 2001, sid 592-611.

[CLA99] D. Clarke, J-E. Elien, C. Ellison, M. Fredette, A. Morcos, R. L. Rivest,"Certficate Chain Discovery in SPKI/SDSI", 1999. http://theory.lcs.mit.edu/~cis/sdsi.html, besökt 6 December 2001.

[COO99] Cooper D A; A Model of Certificate Revocation, 1999. http://csrc.nist.gov/pki/documents/acsac99.pdf, besökt 31 okt 2002.

[DOH02] Dohrmann S, Ellison C; Public-key support for collaborative groups, 2002.

[ELI98] J-E. Elien, "Certificate Discovery Using SPKI/SDSI 2.0 Certificates", examensarbete, 1998. http://theory.lcs.mit.edu/~cis/sdsi.html, besökt 6 December 2001.

[FM99] Försvarsmaktsidé 2020, Rapporterna 1-5, Årsrapporter från perspektivplaneringen Stockholm, Försvarets bok- och blankettförråd.

[FM00] Försvarsplan 2000

[FM01] Försvarsmaktens grundsyn ledning 2001, Stockholm, FM 2001, 28 s.

[FOK02] K. Fokine, Key Management in Ad Hoc Networks, Examensarbete i Informationsteori, Linköpings Tekniska Högskola, LITH-ISY-EX-3322-2002.

[GEN96a] R. Gennaro et al, Robust and Efficient Sharing of RSA Functions, Advances in Cryptology: Proceedings of CRYPTO 96, Springer-Verlag, 1996, sid 157-172.

[GEN96b] R. Gennaro et al, Robust Threshold DSS Signatures, Advances in Cryptology: Proceedings of Eurocrypt 96, Springer-Verlag, 1996, sid 354-371.

[GOL99] Dieter Gollmann, Computer security, John Wiley & Sons, 1999.

[GUI89] J. Guillou, J.-J. Quisquater, “Des Procédés d’Authentification Basés sur une Publication de Problèmes Complexes et Personnalisés don’t les Solutions Maintenues Secrètes Constityent autant d’Accréditations”, Proceedings of SECURICOM ’89, 1999, sid 149-158.

[HOU02] Housley R, Polk W, Ford W & Solo D; Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, 2002. ftp://ftp.rfc-editor.org/in-notes/rfc3280.txt, besökt 31 okt 2002.

[ITS] ITS, Informationstekniska standardiseringen I Sverige, “Terminologi för Informationssäkerhet”, Rapport ITS 6, mars 1994.

[KOH78] Kohnfelder, Loren M: Towards a practical public-key cryptosystem, bachelor’s thesis, MIT, 1978. http://theses.mit.edu/Dienst/UI/2.0/Describe/0018.mit.theses/1978-29, besökt 31 okt 2002.

[KOT85] S.C. Kothari, Generalized Linear Threshold Scheme, Advances in Cryptology: Proceedings of CRYPTO 84, Springer-Verlag, 1985, sid 231-241.

[KRA97] H. Krawczyk, M. Bellare, R. Canetti, "HMAC: Keyed Hashing for Message Authentication", RFC 2104, februari 1997. ftp://ftp.isi.edu/in-notes/rfc2104.txt, besökt 6 December 2001.

[LEI94] T. Leighton, S. Micali, "Secret-key Agreement Without Public-key Cryptography", Advances in Cryptology: Proceedings of CRYPTO 93, Springer-Verlag, 1994.

[MCC00] S. McClure, J. Scambray, "Microsoft wants to be the one company to secure the Internet, but is it right for you?", InfoWorld 17 apr 2000. http://www.infoworld.com/articles/op/xml/00/04/17/000417opswatch.xml, besökt 19 November 2001.

[MCD00] P. McDaniel and A. Rubin, "A Response to `Can We Eliminate Certificate Revocation Lists?' ", Proc. Financial Cryptography 2000, February 2000. http://citeseer.nj.nec.com/mcdaniel00response.html, besökt 31 okt 2002.

[MIC96] Micali S; Efficient Certificate Revocation, 1996. http://citeseer.nj.nec.com/micali96efficient.html, besökt 31 okt 2002.

[MOL93] R.Molva, G.Tsudik, E. Van Herreweghen, S. Zatti, "KryptoKnight Authentication and Key Distribution System", 1993. http://citeseer.nj.nec.com/23399.html, besökt 19 November 2001.

[MYE99] Myers M, Ankney R, Malpani A, Galperin S & Adams C; X.509 Internet Public Key Infrastructure Online Certificate Status Protocol, 1999. ftp://ftp.rfc-editor.org/in-notes/rfc2560.txt, besökt 31 okt 2002.

[MYE01] Myers M, Ankney R, Adams C, Farrel S & Covey C; Online Certificate Status Protocol, version 2, 1999. http://www.ietf.org/proceedings/02mar/I-D/draft-ietf-pkix-ocspv2-02.txt, besökt 31 okt 2002.

[NEU93] C. Neuman, J. Kohl, "The Kerberos Network Authentication Service (V5)", RFC 1510, 1993. ftp://ftp.isi.edu/in-notes/rfc1510.txt, besökt 19 November 2001.

[NEU94] C. Neuman, T. Ts'o, "Kerberos: An Authentication Service for Computer Networks", 1994. http://www.isi.edu/gost/publications/kerberos-neuman-tso.html, besökt 19 November 2001.

[NOA98] Naor M & Nissim K; Certificate Revocation and Certificate Update, published in the Proceedings of the 7th USENIX Security Symposium San Antonio, Texas, January 26-29, 1998. http://www.usenix.org/publications/library/proceedings/sec98/full_papers/nissim/nissim.pdf, besökt 31 okt 2002.

[PFL97] Pfleeger C P; Security in Computing, Prentice-Hall, inc., 1997.

[PGP] http://www.pgpi.org, besökt 30 okt 2002.

[RIV98] R. Rivest, "Can we eliminate certificate revocation lists?", Proceedings of Financial Cryptography '98; Springer Lecture Notes in Computer Science No. 1465 (Rafael Hirschfeld, ed.), Februari 1998, sid 178-183.

[SCH96a] B. Schneier, Applied Cryptography: Protocols Algorithms and Source Code in C, Sec. Ed., John Wiley & Sons, 1996, sid 52-56.

[SCH96b] B. Schneier, Applied Cryptography: Protocols Algorithms and Source Code in C, Sec. Ed., John Wiley & Sons, 1996, sid 71-73, 528-531.

[SDSI] http://theory.lcs.mit.edu/~cis/sdsi.html, besökt 4 nov 2002.

[SIG98] Säkerhetsarkitekturer, Dataföreningen i Sverige, SIG Security, 1998, sid 60-63.

[SPKI] http://world.std.com/~cme/html/spki.html, besökt 4 nov 2002.

[WIL99] Willemson J; Certificate Revocation Paradigms, 1999. http://home.cyber.ee/jan/certif/CRP.ps, besökt 31 okt 2002.

[ZHO99] L. Zhou and Z.J. Haas, Securing Ad Hoc Networks, IEEE Network, Nov/Dec 1999, sid 24-30.