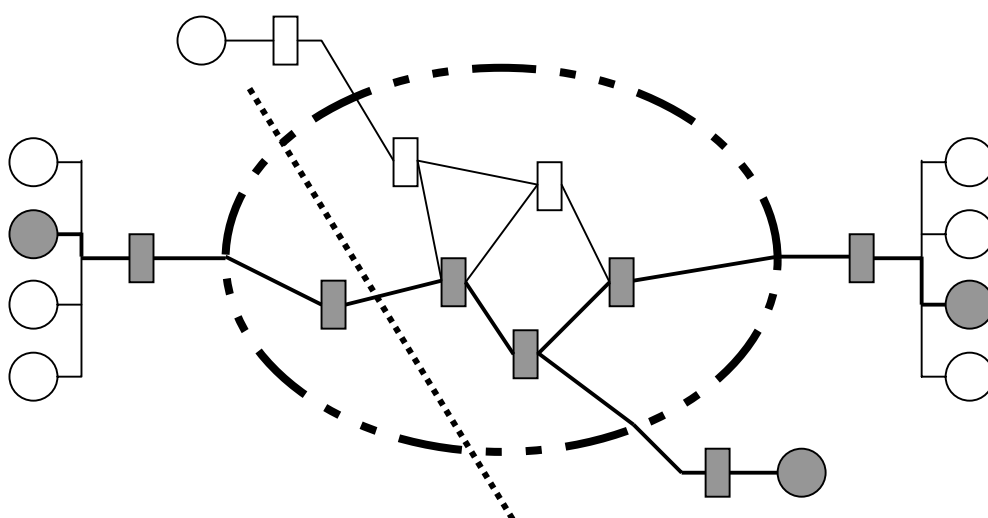


Alf Bengtsson, Amund Hunstad, Lars Westerdahl

Autentisering i nätverksbaserade system



TOTALFÖRSVARETS FORSKNING SINSTITUT

Ledningssystemteknik

Box 1165

581 11 Linköping

FOI-R--0331--SE

December 2001

ISSN 1650-1942

Användarrapport

Alf Bengtsson, Amund Hunstad, Lars Westerdahl

Autentisering i nätverksbaserade system

Utgivare Totalförsvarets Forskningsinstitut - FOI Ledningssystemteknik Box 1165 581 11 Linköping	Rapportnummer, ISRN FOI-R--0331--SE	Klassificering Användarrapport
	Forskningsområde 4. Spaning och ledning	
	Månad, år December 2001	Projektnummer E7023
	Verksamhetsgren 1. Forskning för regeringens behov	
	Delområde 41 Ledning med samband och telekom och IT-	
Författare/redaktör Alf Bengtsson Amund Hunstad Lars Westerdahl	Projektledare Alf Bengtsson	
	Godkänd av Lennart Nyström	
	Uppdragsgivare/kundbeteckning FM	
	Tekniskt och/eller vetenskapligt ansvarig Lennart Nyström	
Rapportens titel Autentisering i nätverksbaserade system		
Sammanfattning (högst 200 ord) Visionen om ett nätverksbaserat försvar medför vissa krav på det kommande ledningssystemet. En lista med sex stycken sådana krav presenteras i rapporten. Utifrån denna kravlista 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 tabell med värdering av de tre autentiseringsklasserna i relation till de sex kraven. De olika metoderna uppfyller kraven på olika sätt. Slutsatsen är att biljett- och certifikatbaserade system är mest generellt användbara, medan identitetsbaserade metoder inte kan skalas upp till stora system.		
Nyckelord autentisering, IT-säkerhet, PKI, nätverksbaserat försvar		
Övriga bibliografiska uppgifter	Språk Svenska	
ISSN 1650-1942	Antal sidor: 43 s.	
Distribution enligt missiv	Pris: Enligt prislista Sekretess	

Issuing organization FOI – Swedish Defence Research Agency Command and Control Warfare Technology P.O. Box 1165 SE-581 11 Linköping	Report number, ISRN FOI-R--0331--SE	Report type User report
	Research area code 4. C4ISR	
	Month year December 2001	Project no. E7023
	Customers code 1. Research for the Government	
	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 Lennart Nyström	
Report title (In translation) Authentication in Network Centric Systems		
Abstract (not more than 200 words) The vision of a network centric defence involves a set of demands for the C² system. In this report six demands are identified. Three classes of methods for authentication - ticket, certificate and identity based respectively - are discussed in relation to the list of demands. 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 six demands. The three classes meet the demands in different ways. Our conclusion is that methods based on tickets and certificates are most general, while identity based methods don't scale well to large systems. .		
Keywords authentication, IT-security, PKI, network centric defence		
Further bibliographic information	Language Swedish	
ISSN 1650-1942	Pages 43 p.	
Price acc. to pricelist Security classification		

Innehåll

1 Inledning	1
2 Sammanfattande slutsatser	3
3 Krav från visionen om ett nätverksbaserat försvar	5
3.1 Kravlista	6
4 Distribuerade informationssystem	9
4.1 Klient/server	12
4.2 Peer-to-peer, P2P	13
4.3 Systemarkitektur i relation till kravlista	14
5 Autentisering	15
5.1 Biljettbaserad autentisering: Kerberos och varianter	16
5.1.1 Varianter av Kerberos	19
5.1.2 Kerberos i relation till kravlista	20
5.2 Certifikatbaserad autentisering: PKI	22
5.2.1 Förvaltning av certifikat: CA och PKI	22
5.2.2 PKI-varianter i relation till kravlistan	24
5.3 Identitetsbaserad autentisering	26
5.3.1 Egenskaper i relation till kravlista	29
6 Metodernas för- och nackdelar i relation till olika försvarsuppgifter	31
6.1 Slutsatser	34
7 Referenser	35

Figurer

Figur 4.1	Mellanskiktet i OSI-modellen	9
Figur 4.2a	Klient/servernätverk	11
Figur 4.2b	Peer-to-peernätverk	11
Figur 5.1	En översikt över Kerberos	17
Figur 5.2	Hierarkier i Kerberos	18

Tabell

Tabell 6.1	Värdering av autentiseringsklasser	34
------------	------------------------------------	----

1 Inledning

Huvuduppgiften avseende ledningsförmåga i det militära försvaret är att kunna leda förband i väpnad strid. Detta ställer krav på tillgången av aktuell information från samtliga aktörer. I de fall då man sätter samman förband av tillfällig art, är det av yttersta vikt att de kan dela information på ett tillförlitligt sätt.

Ett system av denna typ blir givetvis stort, vilket i sin tur gör det lämpligt att dela upp det i hanterbara mindre delsystem, vilka tillsammans bildar ett system av system. En målsättning för systemet är att det inte skall finnas några tekniska begränsningar över hur informationen kan röra sig. Ett plattare informationsflöde snabbar upp beslutsprocessen, då mer information blir tillgänglig.

Tillgängligheten har dock en baksida. Säkerhetskraven på systemet blir höga, då det är av yttersta vikt att rätt information hamnar hos rätt person. En förutsättning för att styra informationen rätt är att systemet känner till vem som har rätt till informationen, vilket i sin tur medför att en användare måste kunna styrka sin identitet inför systemet. Denna funktion benämns autentisering.

För att säkerhetsställa att en aktör är den som han/hon/den/det påstår sig vara, är autentisering i systemet en förutsättning. Om systemet inte kan avgöra vem/vad som efterfrågar en viss information kan det inte heller avgöra om aktören är behörig.

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 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. Kapitel 4 ger en bakgrundsbeskrivning till de arkitekturer som ligger till grund för ett distribuerat system. Här beskrivs kort den välkända klient/server modellen, men även modellen för Peer-to-peer. Tre olika autentiseringsklasser presenteras i kapitel 5 och jämförs i kapitel 6.

2 Sammanfattande slutsatser

Autentisering kan utföras med olika metoder. De klasser av metoder som presenterats och jämförts i denna rapport visar både på för och nackdelar. Även om en klass utmärker sig i vår bedömning bör man ha i åtanke att inom klassen finns flera olika implementeringar, och det är skillnad mellan dem. Det går inte generellt att utse en metod som mest lämplig, utan det kan först ske då förutsättningarna för systemet är klarlagt.

Viktigt att notera är också att det inte går att ta det bästa ur en klass och komplettera en annan klass med. Klasserna bygger på skild teknik och är inte kompatibla.

För att kunna värdera de olika autentiseringsmetodernas för- och nackdelar behövs en detaljerad lista med krav på egenskaper hos ledningssystemet. Vi har formulerat en sådan kravlista, men den är inte tillräckligt komplett för att vi skall kunna göra en definitiv värdering. Även om vi inte kan dra några bestämda slutsatser bedömer vi att biljett- och certifikatmetoder är starka kandidater med utvecklingspotential för ett framtida ledningssystem i ett nätverksbaserat försvar.

3 Krav från visionen om ett nätverksbaserat försvar

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 "nätverkscentrerad krigföring" är inte närmare definierad i [FM99]. Den innebörd vi lägger i termen är 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 krav. Det skall stödja försvarets logistik, det skall kunna samverka med civila system mm.

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 internationella 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.

3.1 Kravlista

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 kravlista senare i rapporten. Som ett allomfattande krav på det framtida ledningssystemet finns "systemet skall vara tillräckligt säkert och robust". I det följande avgränsas aspekter på säkerheten till enbart autentiseringsfunktionen.

- I. 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 Internet-teknik, som på något sätt måste göras "tillräckligt säker". Flexibilitetskrav gör att man skall välja leverantörs- och plattformsoberoende lösningar.
- II. 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.
- III. Verksamheten är organiserad i en befäls- och ansvarshierarki. Tekniken i ledningssystemet skall ändå underlätta informationsutbyte oberoende av nivågränser.
- IV. Informationen skall finnas så nära ägaren (insamlaren) som möjligt. Detta bl.a. för att tydliggöra ansvarsförhållanden.
- V. Delar av systemet skall kunna knoppas av, eller på annat sätt sättas upp, vid uppbyggnad av insatsstyrkor, eller vid andra behov av autonoma delsystem.
- VI. 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.

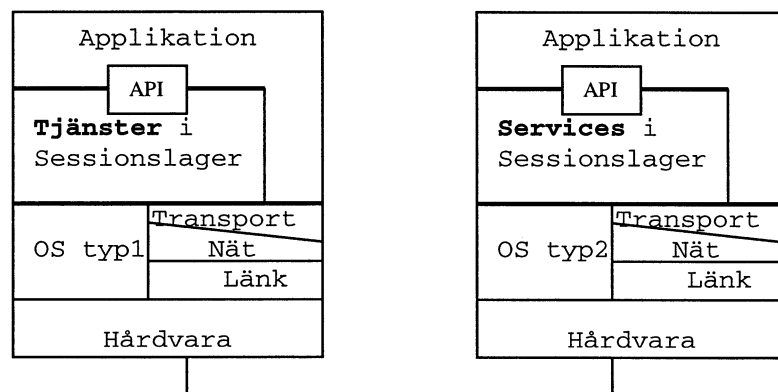
4 Distribuerade informationssystem

En första fråga att ställa sig är vad man menar med ett distribuerat system. Nedanstående kortfattade beskrivning är hämtad ur [BEN00].

Systemet är distribuerat geografiskt, dvs. systemets delar är utspridda på olika platser. Hopkopplingen av delarna sker via ett nät, eller med en svengelsk terminologi ett nätverk (network). Men hopkopplingen kan vara olika omfattande, vilket gör att det totala systemet kan få olika arkitekturer.

En ytterlighet är när systemets delar är så hårt hopkopplade att systemet upplevs som en enda dator. Ett exempel är parallelldatorsystem, där man för att uppnå högsta möjliga prestanda sprider beräkningsdelar för parallell beräkning på de processorer som för tillfället råkar vara lediga. För att realisera detta krävs ett speciellt operativsystem, och ofta också specialskrivna applikationer.

Den andra ytterligheten är ett väldigt löst hopkopplat system, där kanske kommunikationen enbart sker via e-post. De olika delarna utgörs då av självständiga datorer av olika typ och med olika operativsystem. Det behövs ingen programvara utöver standard operativsystem och kommunikationsprogram.



Figur 4.1 Mellanskiktsprogram i OSI-modellen

Det vi här menar med distribuerat system är något slags mellanting mellan ytterligheterna. Man utgår från standard operativsystem och kommunikationsprogram, som skall kunna vara av olika typer. Oberoende av vilken typ av OS som finns i botten skall applikationsprogrammen se ut på ett enhetligt sätt. Mellan operativsystemen, av standardtyp, och applikationen finns därför mellanskiktsprogram (middleware, services, mm) som håller ihop systemet och som gör att användaren inte behöver hålla reda på var olika delar finns. Funktioner som hanteras av mellanskiktet kan vara - adressering av filer och andra objekt, översättning mellan internt format och standardformat, mm. Inte minst kan i mellanskiktet finnas olika säkerhetsfunktioner - nyckelhantering, inloggning, grupp tillhörighet mm. Om man ritar in mellanskiktet i OSI-modellen Open Systems Interconnect [SIG98], hamnar det i sessionslagret ungefär enligt figur 4.1. För en omfattande diskussion av denna arkitektur hänvisas till [FJÄ00].

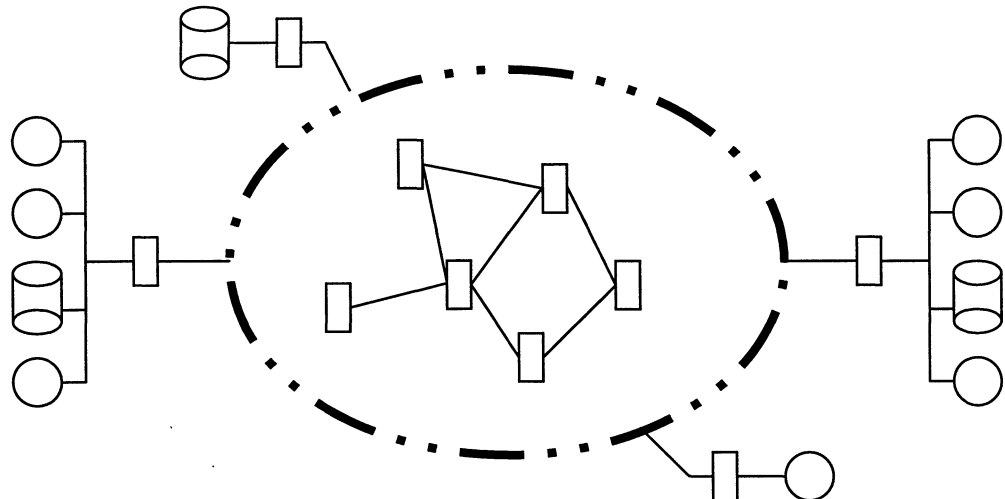
Figur 4.1 illustrerar att mellanskiktet logiskt sett ligger mellan applikationerna och operativsystemet. Det finns ett API, Application Program Interface, som talar om hur applikationerna anropar de olika funktionerna i mellanskiktet. Ju fler funktioner som finns desto mer komplicerat blir naturligtvis API. När en applikation skall anpassas till en distribuerad miljö måste den kompletteras med API-anrop. Om det är en komplicerad arkitektur, och särskilt om applikationen är skriven i ett ovanligt språk, kan detta innebära ett mycket stort arbete. Eftersom det finns ett antal alternativa arkitekturer, blir följden att de flesta applikationer inte har anpassats till alla arkitekturer.

En idag vanlig arkitektur mellan ytterlighetsnivåerna är klient/server arkitekturen. På klienten körs huvuddelen av applikationen. Då och då skickar klienten olika typer av begäran (requests), t.ex. hämta/lagra data, till en eller flera servrar. Servern skickar tillbaka ett svar eller en kvittens varefter klienten fortsätter sin avbrutna applikation. Ibland menar man också med klient/server att det skall vara tillståndslöst, dvs. att varje fråga/svar skall vara oberoende av andra frågor/svar. I så fall tillåter man t.ex. inte att man vid första fråga/svar utför speciella kontroller som man sedan litar på i fortsättningen.

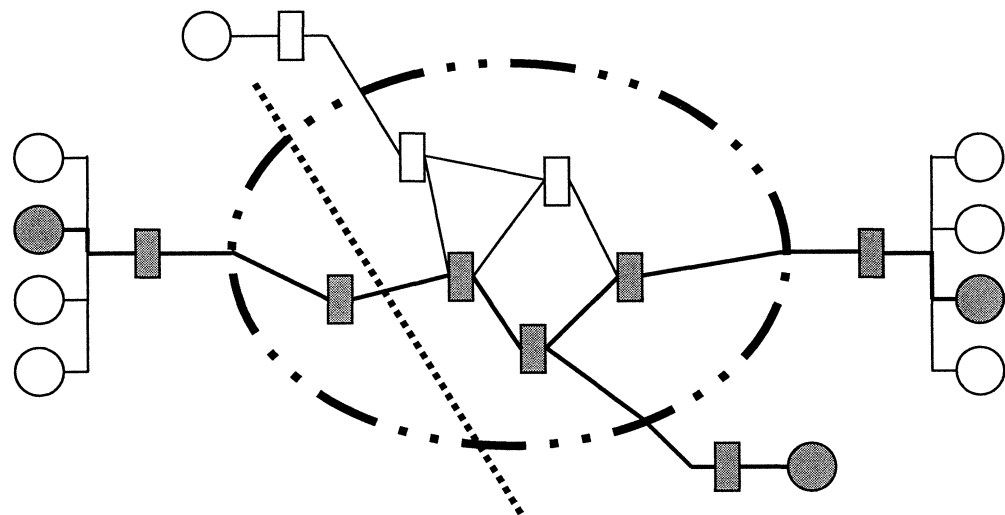
Beroende på arkitekturen hos det distribuerade systemet finns olika många tjänster, dvs. olika mycket funktionalitet, i mellanskiktet. Ett exempel på en löst kopplad arkitektur är WWW. I det enklaste fallet finns i stort sett bara http-protokollet, för att överföra data till/från en WWW-server, i mellanskiktet. För att erhålla bättre säkerhet kan flera tjänster (services) läggas i mellanskiktet, jämför TLS [ALL99]. Ett exempel på en relativt hårt kopplad arkitektur är DCE, Distributed Computing Environment [HAR92]. I DCE finns mängder med funktioner i mellanskiktet.

Ju större och mer distribuerat ett system är desto större blir komplexiteten och därmed säkerhetsproblemen. Svåra delproblem inom autentiseringsfunktionen är bl.a. administration och distribution av kryptonycklar. Om sådana funktioner skall kunna hanteras centraliserat, vilket har många fördelar, krävs ett ganska hårt kopplat system. Å andra sidan blir väl konstruerade löst kopplade system mer flexibla och robusta.

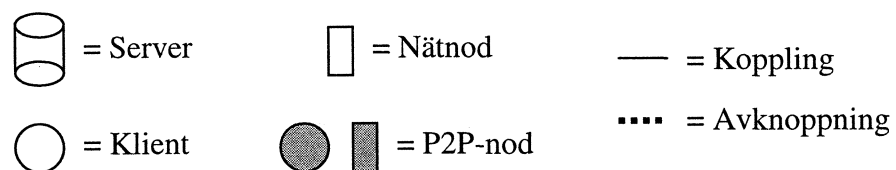
Figur 4.1 åskådliggör hur programvaran för olika tjänster kan ligga som komponenter i ett mellanskikt, mellan applikationer och operativsystem, inuti datorerna. Figur 4.2 åskådliggör hur datorerna och andra typer av noder i ett distribuerat system, är hopkopplade med varandra i klient/server- respektive peer-to-peer-lösningar, vilka beskrivs närmare i kapitel 4.1 och 4.2.



Figur 4.2a Klient/servernätverk



Figur 4.2b Peer-to-peernätverk



Det i särklass största och viktigaste distribuerade system som finns är det globala Internet. Krav (I) på det framtida ledningssystemet, det skall baseras på civil teknik, gör att utvecklingstendenser på Internet är viktiga att följa.

Fram till slutet av 1980-talet utvecklades Internet i rask, om än inte explosionsartad, takt till ett allt större och kraftfullare kommunikationsnät. Användarna letade upp varandra och utväxlade data sins emellan. Data var lagrade ute hos användarna och en användare var ibland sändare och ibland mottagare i dataöverföringarna. I och med tillkomsten av WWW, World Wide Web, med standarderna HTTP och HTML, i början av 90-talet ändrades användningen av Internet ganska

radikalt. Internet är nu i huvudsak ett gigantiskt informationssystem med klient/server-arkitektur.

4.1 Klient/server

Klient/server-arkitekturen kännetecknas av att all information finns lagrad centraliserat. Detta gäller såväl själva data som olika former av metadata, dvs. information om data. Speciellt viktiga metadata är olika former av adresser, pekare och index som används för att hitta alla objekt och all information som finns på Internet. När en klient (en användare eller hans applikationsprogram) hämtar data från en server görs detta ofta i flera steg. Först begär klienten metadata från en metadataserver. Detta kan behöva upprepas med begäran om nya metadata från en annan server. Slutligen begär klienten data från den datalagrande servern. Liknande procedur genomlöps när en klient vill lagra data i en server, fast begäran att få hämta ersätts naturligtvis då med begäran att få lagra.

Ett exempel på server för metadata i form av adresser är Domain Name Server, DNS. En DNS svarar i princip på en begäran av typ "jag vill veta den numeriska IP-adressen för datorn med namnet *datorNN.foi.se*". Det bör påpekas att alla adresser inte finns i en och samma server, utan DNS-servrarna samarbetar. Om en server inte kan svara på frågan slussas den vidare till en annan server, tills så småningom begäran kan besvaras.

Ett exempel på server för pekare, eller om man vill kalla dem för logiska adresser, är PURL, Persistent Uniform Resource Locator [PURL]. När ett clientprogram vill hämta data från en WWW-server skickas en hämtningsbegäran med en pekare, URL, som kan se ut t.ex. som

www.foi.se/intern_katalog1/intern_katalog2/data. Då anropas först DNS-servern, för att få den numeriska IP-adressen till WWW-servern *www.foi.se*. Sedan anropas WWW-servern, som letar fram data i den interna filstrukturen. Om data nyligen har flyttats till en annan server, eller till annan plats, kommer hämtningen att misslyckas. Det finns därför behov av att peka på data på ett mera beständigt sätt, utan namn på interna datorer och kataloger. Försök pågår att standardisera Uniform Resource Name, URN. Eftersom DNS är en väletablerad och standardiserad funktion vore det naturligt att utvidga DNS så att denna kan svara på frågan "var finns just nu resursen URN?". Men just för att DNS är så etablerad, och kritisk för all kommunikation på Internet, så vill man inte gärna ändra DNS. Det är heller inte lätt att komma fram till det bästa utseendet på URN. I förlängningen kan det liknas vid att klassificera all information på ett enhetligt sätt.

Ett exempel på servrar för metadata i form av index är de olika sökmotorernas index över ordförekomsten i olika typer av datafiler. En mycket viktig form av metadata är den som används för att autentisera användare och enheter i ett system. Vilken form dessa metadata tar beror på autentiseringsmetoden, exempelvis biljett- eller certifikatsystem.

Klient/server-arkitekturen har många fördelar. Det är en etablerad teknik, med många standardprodukter. Det är alltid lättare, sett ur administratörens perspektiv, att hantera data som är centraliserade än om de är utspridda på många ställen. Speciellt gäller detta för metadata, t.ex. namn och adresser. Det är ju väl känt att det är lättare att upprätthålla datakonsistens och att rätta fel om data är centralise-

rade. Vidare är det relativt lätt att bygga ut systemet med nya funktioner. Det kanske bara behövs några nya servrar med data och tillhörande metadata.

Klient/server-arkitekturen har också nackdelar. En centraliserad server blir en kritisk punkt. Fallerar den, eller störs ut, kan hela systemet raderas. Visserligen brukar man dubblera servrar och servrar brukar samarbeta. Men det är likväl stor risk att nödvändiga data är otillgängliga om en del av systemet skall klara sig autonomt. Klient/server-system är inte självorganiserande. Vidare lagras data långt från ursprungskällan, vilket kan försvåra upptäckt av fel.

En genomgång av kravlistan (I-VI) ger vid handen att klient/server-arkitekturen har fördelar vad gäller (I) civil teknik, (II) informationshantering, (VI) system-av-system. Däremot finns nackdelar vad gäller (III) oberoende av hierarki, (IV) data hos ägaren, (V) autonomitet och självorganisering.

4.2 Peer-to-peer, P2P

På senare år har det på Internet växt fram andra tekniker som komplement till klient/server-arkitekturen. På sätt och vis kan man säga att det är en återgång till det gamla, en återgång till tiden före WWW. Men skillnaden är att det inte bara är två slutapplikationer, sändare och mottagare, som kontakter varandra för att utväxla data. Det är i stället en grupp av slutapplikationer som på något sätt vill dela resurser med varandra. Dessa slutapplikationer bildar ett självorganiserat nät, ett ad-hocnät. Termen som myntats för detta är peer-to-peer arkitektur, P2P [OPEN, PAR01]. Begreppet peer-to-peer skall markera att nätet bildas mellan likvärdiga parter utan insats av speciellt kraftfulla noder, t.ex. servrar.

Det är framför allt två tillämpningar som drivit fram P2P. Den ena tillämpningen är utdelning av outnyttjade beräkningsresurser, för att kunna hjälpas åt med tunga beräkningar. Ett exempel är seti@home för att leta efter intelligent liv genom signalanalys av signaler från rymdteleskop. Den andra stora tillämpningen har varit fildelning, framför allt delning av musik i mp3-format. De mest kända tillämpningarna är Napster och Gnutella.

Grovt sett kan etableringen av P2P-nät delas in i två steg

- a) En likvärdig aktör (peer) skall leta upp sina motparter i gruppen.
- b) De önskade resurserna skall delas, t.ex. data skall överföras

Det första steget är det som är svårast att genomföra utan tillgång till centralt lagrade metadata. T.ex. Napster använder en central server med adresser (metadata), medan data, mp3-filerna, ligger decentraliserat hos användarna. Napster sägs därför ibland inte vara "äkta" P2P. Däremot klarar sig Gnutella utan adressserver.

För- och nackdelar med P2P-arkitektur är i stort sett spegelvänt i förhållande till klient/server-arkitektur. Fördelarna är framför allt att nätet är självorganiserande, utan sårbara, centrala noder, och att data stannar nära källan.

Nackdelarna är svårigheten att hålla konsistenta och felfria data. Inte minst gäller detta metadata i steg a) ovan. Om man inte har korrekt autentiserade identiteter

blir ett P2P-nät för fildelning ett eldorado för spridning av falska data, t.ex. virus. Det kan också vara svårt att utöka med nya funktioner i och med att varje peer måste "lära sig" det nya tillägget.

En genomgång av kravlistan (I-VI) ger att P2P har fördelar vad gäller (III) oberoende av hierarki, (IV) data hos ägaren, (V) autonomitet och självorganisering. Däremot finns nackdelar vad gäller (I) civil teknik, (II) informationshantering och (VI) system-av-system.

4.3 Systemarkitektur i relation till kravlista

Ovanstående avsnitt om klient/server respektive P2P avslutas med kortfattade kommentarer om hur respektive arkitektur relateras till kravlistan i kapitel 3. Denna jämförelse är mycket summarisk. Den säger i stort sett att de båda arkitekturerna är varandras spegelbild i relation till kravlistan. Det som avgör respektive arkitekturs egenskaper är framför allt i hur hög grad informationen lagras centraliserat, respektive distribuerat.

Syftet med denna rapport har inte varit att försöka värdera hela systemarkitekturen, syftet har varit att värdera en del av arkitekturen, autentiseringsfunktionen. Därför kommenterar vi här inte ytterligare systemarkitekturen i relation till kravlistan. I följande kapitel 6 görs däremot en sådan värdering av autentiseringsmetoder i relation till kravlistan.

5 Autentisering

Autentisering betyder äkthetsbekräftelse. Metoder för autentisering betyder alltså metoder att verifiera att någonting är äkta. Nästa fråga blir då; vad menar man med "någonting" och vad innebär "äkta"?

Beträffande "någonting", autentiseringsobjektet, menar man ibland passiva objekt, t.ex. olika typer av digitala data. Det kan vara datapaket, datafiler, digitala bilder mm. För att tydliggöra att man menar passiva data använder man ibland begrepp som dataautentisering eller dataintegritet. Begreppet autentisering isolerat, till skillnad från det sammansatta dataautentisering, brukar avse aktiva aktörer. Det kan vara mänskliga användare, datorer, kommunikationsutrustning, dataprogram mm.

Innebörden av "äkta" kan vara att man menar att objektet inte, medvetet eller omedvetet, har förvanskats över tiden. Det vill säga att objektet till 100 % stämmer överens med ett ursprungligt original. Med "äkta" kan man också mena att en aktör inte agerar i någon annans namn, dvs. att en uppgiven identitet inte är falsk.

I denna rapport innebär autentisering 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. datorer och kommunikationsutrustning. Matchning av fingeravtryck och andra "analoga metoder" att identifiera människor avhandlas alltså inte.

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. 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. I annat fall kan man inte koppla ihop delsystem, och det blir svårt att skala upp till ett totalt, stort system.

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 fem andra frågeställningar i denna rapport

- (a) Hur löser man grundbulten - autentiseringsnyckeln skall hemlighållas för alla utom ägaren (ägarna), i tillägg till den nämnda möjligheten att baklänges räkna fram autentiseringsnyckeln.
- (b) Hur visar man, på ett säkert sätt, att man besitter nyckeln? 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 ingå. Hela detta steg kallas autentiseringsprotokoll, se vidare [SCH96a].
- (c) Hur knyter man, på ett säkert sätt, ihop en aktörs verifieringsnyckel med hans identitet (annars kan han uppträda under falskt namn)?
- (d) Hur hanteras födelsetprocessen? Kan denna delegeras, så att delsystem kan kopplas samman?
- (e) Hur löses problemet att en aktör skall kunna agera under olika identiteter (t.ex. olika namngivningssätt när två delsystem skall kopplas samman, en aktör i olika roller mm)?

Dessa fem frågeställningar är väsentliga att lösa i det komplexa system-av-system som beskrevs i avsnittet om distribuerade system. Denna rapport handlar om några tänkbara ansatser i tre olika klasser av autentiseringsmetoder. Dessa klasser kommenteras i relation till kravlistan I-VI. En grov värdering av klassernas för- och nackdelar görs i form av graderingen ++, +, 0, -, --.

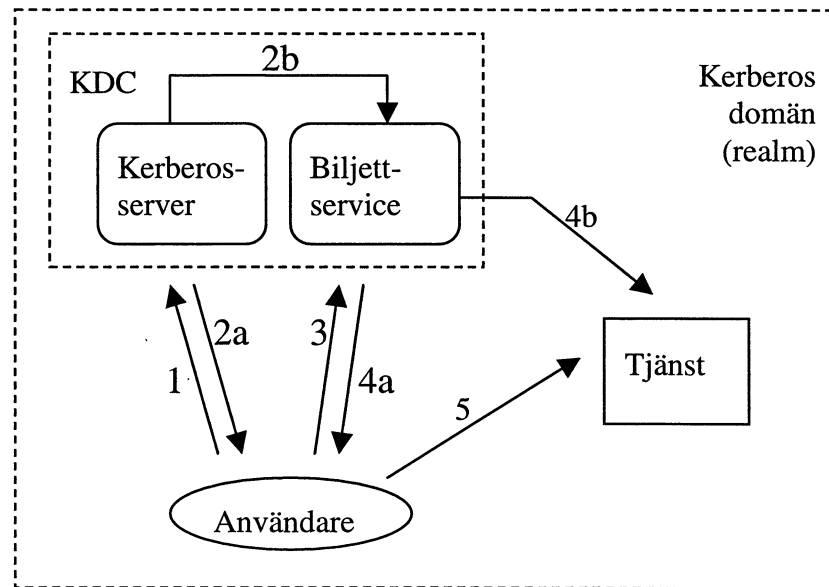
5.1 Biljettbaserad autentisering: Kerberos och varianter

Kerberos är ett etablerat autentiseringssystem vilket har utvecklats sedan mitten av 1980-talet. Det var ursprungligen inte ämnat som ett allmänt system, utan som en del av projektet Athena, ett lokalt nätverk på MIT. Först version 4 av Kerberos spreds utanför MIT. Den senaste versionen, version 5 från 1993, är den som beskrivs i detta kapitel. I och med att systemet har några år på nacken och därmed analyserats flitigt görs här endast en översiktlig genomgång. För en detaljerad beskrivning av Kerberos version 5 se RFC 1510 [NEU93]. Version 5 har förbättrats väsentligt jämfört med version 4, främst genom införandet av tidsangivelser (timestamps).

Autentiseringsprocessen sköts genom delegering, dvs. användaren lämnar över ansvaret till Kerberosservern. Systemet har på egen hand möjlighet att verifiera en användares behörighet. En positiv effekt av Kerberosystemet är att inga lösenord skickas över nätverket, vare sig klartext eller i krypterad form.

En förutsättning för systemets tillförlitlighet är att den initiala autentiseringen, dvs. då användaren för första gången registreras i Kerberosserverns accesslista, sker på ett säkert sätt (steg (d) ovan). Vanligtvis sker detta hos en systemadministratör samtidigt som användaridentitet och lösenord delas ut, då användaren introduceras i systemet. Administratören har då möjlighet att verifiera personens identitet och rättigheter via körkort, arbetsuppgifter mm [NEU94].

En enkel översikt av hur Kerberos fungerar beskrivs i figur 5.1.



Figur 5.1 En översikt över Kerberos

1. När en användare loggar in skickas användaridentiteten samt namnet på aktuell biljettservice till Kerberosservern.
2. Kerberosservern verifierar att användaren är med på listan över auktoriserade användare samt:
 - a. Skickar en tillfällig nyckel (session key) som skall användas för kommunikation med biljettservicen samt en biljett till biljettservicen. Den tillfälliga nyckeln krypteras med användarens hemliga nyckel.
 - b. Skickar en kopia av den tillfälliga nyckeln samt identiteten på den användare som skall använda nyckeln.
3. Användaren skickar sin nyligen erhållna biljett samt en begäran om en ny biljett till en önskad tjänst, krypterad med sin tillfälliga nyckel.
4. Biljettservicen kan nu bekräfta att användaren är auktoriserad genom att jämföra användarens information med den skapad av Kerberosservern.
 - a. Stämmer detta överens, returnerar biljettservicen en biljett till aktuell tjänst och en tillfällig nyckel att använda för kommunikation med servern.
 - b. Biljettservicen skickar även en kopia av biljetten till aktuell server.
5. Användaren kan nu, genom att använda sin biljett och krypteringsnyckel, hämta önskad tjänst.

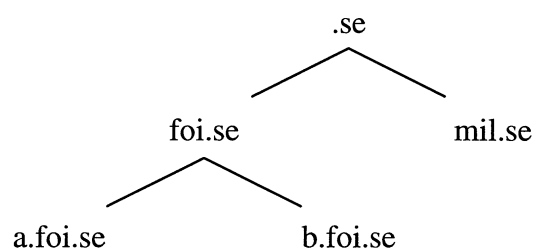
I praktiken är steg 2 – 5 dolt för användaren. Autentiseringen mot Kerberos-servern sker en gång, i samband med inloggning på arbetsstationen. En vanlig term för detta är *Single Sign-On*, vilket innebär att ett lösenord endast skall behövas skrivas in en gång. Kommunikationen med biljettservicen sker varje gång en användare begär en tjänst. Det är först om autentiseringen skulle misslyckas som en användare får en notis av systemet.

Den biljett som erhålls av biljettservicen är inte bara ett bevis på att användaren har blivit godkänd av Kerberosservern, den innehåller även vissa begränsningar för användaren. Dels har biljetten ett utgångsdatum för att hindra återuppspelning, dels finns viss auktoriseringsinformation, även om Kerberos inte kan räknas som en fullfjädrad auktoriseringstjänst.

I Kerberos används nycklar med två olika beständigheter. Mellan användare och Kerberosystemet skapas sessionsnycklar, dvs. engångsnycklar vilka inte återanvänds. De nycklar som brukas mellan Kerberosenheter, inom KDC:n och mellan KDC:n och aktuell tjänst, återanvänds.

Vanligtvis är Kerberosservern och biljettservicen endast logiskt åtskilda. De ligger ofta på samma fysiska server, vilken brukar förkortas KDC (key distribution center). En KDC kan innehålla flera biljettserviceenheter. Vidare samlas KDC:n tillsammans med en uppsättning klienter (användare) och applikationsservrar i en administrativ domän kallad realm. Inom en Kerberos domän litar alla klienter och servrar blint på KDC:n.

Det kan hända att en användare vill ha tillgång till en tjänst utanför den egna domänen. Autentiseringsmetoden för detta är den samma som innan, med tillägget att den egna servern måste dela en hemlig nyckel med aktuell biljettservice i den andra Kerberos domänen. Version 5 av Kerberos stödjer flerstegskommunikation (multi-hop) [NEU94], vilket ger möjlighet till en hierarkisk nyckeldistribution. I praktiken betyder det att varje domän delar nycklar med sin förälder och sina barn (figur 5.2).



Figur 5.2 Hierarkier i Kerberos

foi.se delar en nyckel med .se, vilken i sin tur delar en nyckel med mil.se. Genom denna indelning kan till exempel a.foi.se hämta en fil hos mil.se med hjälp av Kerberosystemet. Om det inte uppstår några problem kommer användaren inte att märka att den egna domänen har lämnats.

En förutsättning för systemet är en kontinuerlig uppkoppling, dvs. en användare måste alltid ha möjlighet att kontakta Kerberosservern. Skulle denna kontakt bry-

tas kan möjligheten att få biljetter drastiskt minska och därmed kan systemet sättas ur funktion. Version 5 av Kerberos löser detta med den så kallade flerstegsfunktionen. När en domän har varit i kontakt med en annan finns det möjlighet att utväxla nycklar så att en direktkontakt kan etableras i fortsättningen istället för att vandra genom hela trädet. Har en sådan utväxling skett, kan systemet överleva trots att en förälder inte är ansluten.

Kerberos största svaghet är lösenorden. Används ett svagt lösenord kan det lätt forceras, vilket sätter hela systemet ur spel. Ersätter man ett mänskligt genererat lösenord med till exempel smarta kort, publika nycklar eller biometriska nycklar tryggas systemet betydligt.

5.1.1 Varianter av Kerberos

Kerberos är en etablerad produkt med några år på nacken. Dess acceptans har lett till att flera varianter uppkommit, främst för att komplettera de svagheter som finns i systemet.

Kerberos i MS Windows 2000

Microsoft har implementerat Kerberos som en standard för autentisering i Windows 2000. Implementeringen följer i stort RFC 1510, även om de har utökat protokollet något. Till exempel använder de sig av ett datafält vilket innehåller specifik information för Windows autentiseringsdata. Fältet, vilket specificerar klientens rättigheter, fungerar bra i ett homogent system men kan komplicera processen för system som inte är baserade på Windows 2000. En biljett från en Unix Kerberos domänen kan mycket väl bli nekad access beroende på att Windows 2000 domänen inte kan avgöra vilka rättigheter klienten skall tilldelas. Datafältet finns visserligen angivet i RFC 1510, men det råder en diskussion om vilket dess syfte egentligen är.

Äldre operativsystem från Microsoft kan inte logga in i en Windows 2000 Kerberos domän, då de saknar nödvändiga säkerhetsfunktioner [MCC00].

Microsoft har även modifierat protokollet genom att använda sig av asymmetriska nycklar (public key, steg 2 i figur 5.1) under inloggningen. Detta ökar säkerheten under processen mest kritiska ögonblick. I övrigt används symmetriska nycklar i och med att det är betydligt snabbare.

Under årens lopp har en del varianter av Kerberos introducerats. Dessa är utvecklingar av Kerberosprotokollet, främst genom modifiering av autentiseringsservern och nyckelhanteringen.

KryptoKnight

När IBM designade KryptoKnight tog de tillvara på de positiva erfarenheter som fanns från Kerberos [MOL93]. Utvecklarna fokuserade på att minska antalet meddelanden som måste skickas över nätverket, längden på meddelandena samt graden av kryptering [SCH96b]. Det är dessa funktioner som är tidskrävande i Kerberos. Lika så har man i hög grad ersatt tidsangivelserna (timestamps) då dessa kräver en synkroniserad klocka.

KryptoKnight erbjuder följande service [MOL93]:

- Single Sign-on: Fungerar i princip som Kerberos.
- Tvåpartsautentisering: När den initiala autentiseringen är genomförd kan en klient etablera direktkontakt med en annan klient eller server. Man har vid varje tillfälle möjlighet att välja om man vill ha envägs- eller tvåvägsautentisering.
- Nyckeldistribution: För att tvåpartsautentisering skall kunna genomföras krävs att de båda enheterna delar en hemlig nyckel. Distributionen av nycklar sker dolt för användaren via autentiseringsservern. När en uppsättning nycklar erhållits lagras de på ett transparant sätt för senare bruk.
- Autentisering av ursprung och innehåll: Under den tid en nyckel är giltig (sessionen) har de båda parterna möjlighet att verifiera ursprung och innehåll på de data som skickas mellan de båda.

Distributed Computing Environment (DCE)

DCE [HAR92] säkerhetsfunktioner omfattar autentisering, auktorisering och accesskontroll. Det ursprungliga DCE protokollet var utvecklat från Kerberos förutsättningar, men använder sig idag av publika nycklar.

En markant skillnad är att DCE:s biljett innehåller en användares privilegier, ungefär som Microsofts Windows 2000 implementation.

5.1.2 Kerberos i relation till kravlista

Kerberos är ett system för autentisering av användare. Systemet använder sig av symmetriska nycklar för en säker kommunikation mellan olika enheter inom systemet. En användares lösenord skickas aldrig över nätet, inte ens i krypterad form.

Det finns möjlighet att bygga ut systemet så att det stödjer asymmetriska nycklar eller engångskoder. En sådan lösning skulle avhjälpa Kerberos största svaghet, nämligen känsligheten för svaga lösenord.

Microsoft har implementerat Kerberos som standard autentiseringssystem i Windows 2000. Deras version stödjer asymmetriska nycklar. Dock finns vissa kompatibilitetsproblem med andra, icke Windows 2000, system. Dessa problem är dock lösbara.

Modifierade Kerberosystem finns på marknaden. I vilket avseende de är modifierade varierar, men i huvudsak rör det sig om vilka och hur nycklar hanteras.

Utgår man ifrån den kravlista som tidigare presenterades i kapitel 3.1 har Kerberos följande egenskaper:

I. Civil teknik:

Kerberos, likt andra serversystem, är civil teknik vilken kan göras plattformsoberoende. Det finns ett antal kommersiella implementeringar av systemet i sig, samt en version som följer med Microsoft Windows 2000. (++)

II. Information till rätt plats:

Kerberos är i huvudsak ett autentiseringsverktyg. Dock är Kerberos utrustad med en begränsad auktoriseringsfunktionalitet. Denna bör i huvudsak ses som en informationsbärare åt en separat auktoriseringsservice. (+)

Vitalt för tillförlitligheten i systemet är att uppkopplingen säkras (fysisk säkerhet) och att nycklar fördelas på ett tillförlitligt sätt. (-)

III. Ansvarshierarki, men ändå icke-hierarkiskt informationsflöde:

Kerberos domäner är hierarkiskt sammankopplade för att säkra nyckeldistributionen. Så länge nycklar är fördelade enligt förälderbarn principen kan information flöda genom hierarkin utan mänsklig inblandning. (+)

IV. Information nära ägaren:

Kerberos är ett klient-serverbaserat system vilket innebär att all data och tjänster är centraliserade. Om man skulle vilja lagra data lokalt (på en arbetsstation) men ändå göra den tillgänglig, krävs att arbetsstationen registrerar sig som en tjänst hos biljettservicen. Generellt finns ägaren och informationen inom samma domän. (-)

V. Små delsystem:

Att skapa ett nytt delsystem inom en befintlig hierarki av klienter och servrar är fullt möjligt, men arbetskrävande. Att förändra antalet klienter inom en domän medför inga speciella problem. Nya tjänster måste dock anpassas ("kerboniseras"). Vill man bygga ut domänen med ännu en biljettservice, omfattar detta mycket administrativt arbete.

Att skapa ett nytt, helt autonomt delsystem innebär att man skapar en helt ny Kerberosmiljö. Detta kan medföra en betydande mängd arbete. (--)

VI. System av system:

"Original" Kerberos definierar hur gränssnitt skall se ut, men en viss anpassning kan behövas när kommersiella system skall samarbeta. (++)

5.2 Certifikatbaserad autentisering: PKI

Resonemanget i detta kapitels inledning redovisade generellt terminologi och tillvägagångssätt vid autentisering. Här klargörs endast vissa ytterligare detaljer. Detta inledande resonemang om publika nyckelsystem utgår i hög grad från Gollmann [GOL99] och svensk terminologi ur ITS 6 [ITS].

Autentisering med publika nycklar baserar sig på asymmetriska krypteringsmetoder, det vill säga att signerings- och verifieringsnycklarna är olika. Signeringsnyckeln är hemlig och verifieringsnyckeln publik. Tankegången här innebär att endast en korrekt digital aktör skall kunna signera, men att alla skall kunna verifiera att signeringen är korrekt. Därigenom kan meddelandets äkthet, innehåll, och avsändarens identitet kontrolleras.

Symmetriska metoder medför betydande förvaltning, för lagring och kommunikation via säkra kanaler, av alla de hemliga nycklarna. Publika asymmetriska nyckelsystem minskar förvaltningsbehoven, just genom att en av nycklarna är öppet tillgänglig. Förvaltningsbehoven är dock inte eliminerade. Kopplingen mellan publik nyckel och olika accessrättigheter respektive mellan signerande användare och hemlig nyckel måste fortfarande säkras. För att realisera detta utges så kallade certifikat vilka utfärdas av en certifieringsmyndighet¹. Den infrastruktur som man därmed bygger upp kring certifieringsmyndigheten benämns i litteraturen med förkortningen PKI (= Public Key Infrastructure).

Det bör omnämnas att i Handbok Totalförsvarets Signalskyddstjänst [TST01] redovisas hanteringen av TAK, Totalförsvarets Aktiva Kort. TAK är RSA-baserad och tänkt att användas till olika militära system för att möjliggöra stark autentisering, digital signering och att vara databärare (företrädesvis nyckelinformation). Handboken specificerar bland annat hantering av spärllistor, kortbytesrutiner och vilken certifikatinformation kortet tillhandahåller.

5.2.1 Förvaltning av certifikat: CA och PKI

En betydande diskussion och forskning fokuserar på hur certifikat och infrastrukturen de ingår i skall designas. Av de viktigaste designfrågorna och –avvägningarna finns bland annat följande:

- Vilken information skall certifikatet innehålla, vilket exempelvis kan vara
 - Namn
 - Verifieringsnyckel
 - Certifieringsmyndighetens namn
 - Certifikatets giltighetsperiod
 - Accessrättigheter

¹ Benämns ofta i litteraturen som CA (Certification Authority).

- Koppling mellan namn, digital aktör och accessrättigheter
- Komplexitet i infrastrukturen relativt säkerhet och smidig användning, särskilt med avseende på uppbyggnad av och hantering av certifikatkedjorna.

Om man kort sammanfattar diskussionen kan man observera, som exempelvis [GOL99] gör, att det finns ett spektrum av PKI-lösningar. I den enda änden av spektrum befinner sig X.509 med en global certifikatbas, vilket åtminstone i sin ursprungliga variant riskerar att medföra, enligt bland annat [CLA99], en komplex lösning med begränsad flexibilitet och prestanda. I den andra änden av spektrum finns PGP² 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]. [CLA99] utgör en utmärkt sammanfattning av SPKI/SDSI-utvecklingen. SPKI/SDSI baseras, i motsats till X.509s globala bas, på en decentraliserad certifikatbas. Varje ägare av en publik nyckel kan generera en lokal certifikatbas som är relaterad till denna nyckel. Ett antal sådana certifikatbaser kan sammanlänkas till en, enligt [CLA99], flexibel och kraftfull PKI-lösning.

SPKI/SDSI förmår, enligt [CLA99], att effektivt hitta korrekt certifikatkedja som visar att aktörens publika nyckel har nödvändiga rättigheter knutna till sig. För att lyckas med detta baserar denna PKI sig på

- Flexibel namngivning
- Systemets förmåga att hantera delegering
- Hantering av ”tröskelsubjekt”, där signering av flera aktörer krävs för åtkomst till skyddad resurs
- Beräkningsmässig effektivitet

SPKI/SDSI har två typer av certifikat, namncertifikat och auktorisationscertifikat. Namngivnings- och auktoriseringsförfarandet är, enligt [CLA99], ortogonal och konceptuellt separerbara. 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, till exempel A, B, Alice eller Bob.
- Subjektet S specificerar aktören närmare, till exempel *Alice Ted*, vilket är en angivelse av ”Ted med relation till Alice” eller motsvarande.

² PGP = Pretty Good Privacy

- V är en validitets-/giltighetsspecifikation. Anges normalt som ett tidsintervall.

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 $v(S)$ är de som mottar auktorisation.
- Delegationsbiten d, om $d = \text{Sann}$, ger varje nyckel i $v(S)$ 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.

Värt att observera är här att lokala namnbaser är knutna till respektive publik nyckel. Med andra ord, som bland annat [ELI98] noterar, ges auktorisation från en publik nyckel till en annan publik nyckel. Auktorisation och namngivning är separerade, och därmed undviks en uppsättning problem man annars får, enligt [ELI98], mellan auktorisering (att delegera rättigheter till något subjekt) och namngivning (bindningen av något subjekt till en identifierare).

Varje K som innehar ett auktorisationscertifikat blir i SPKI/SDSI en certifieringsmyndighet. Delegeringsstrukturen, som är inbyggd i SPKI/SDSI, bygger därmed upp en hierarki av certifieringsmyndigheter.

I namngivningsförfarandet i SPKI/SDSI ligger också möjligheterna för, som [CLA99] bedömer det, en flexibilitet som även medför beräkningsmässiga fördelar vid hantering av certifikatkedjor. Systemet baserar sig i stor grad på så kallade omskrivningsregler och tillhörande beräkningar för att transformera ett certifikat (eller sträng) till ett annat. På liknande sätt kan två separata certifikat reduceras till ett.

Separata spärrlistor existerar inte i SPKI/SDSI. Subjektet S i relevant namncertifikat tillsammans med validiteten V erbjuds istället. [RIV98] redovisar argument för denna lösning och varför spärrlistor i de flesta fall inte är en bra lösning.

5.2.2 PKI-varianter i relation till kravlistan

Om man relaterar diskussionen om autentisering med publika nycklar och PKI till kravlistan i kapitel 3.1, kan man göra vissa objektiva observationer och andra mera värderande observationer. Kravlistan nödvändiggör viss fortlöpande tolkning, exempelvis av vad som bör läggas i ett begrepp som "tillräckligt säker".

I. Civil teknik:

Civil teknik och civila system bedömer vi kan användas, i de fall där de uppfyller specificerade säkerhets- och funktionskrav. Exempelvis torde PGP i försvarssammanhang oftast inte vara en lämplig lösning. De allra flesta försvarstillämpningar kräver striktare certifiering än att vara bekant till någon annan i PGP-kedjan.

SPKI/SDSI bedömer vi inneha en intressant utvecklingspotential. Det är dock nödvändigt att göra en närmare bedömning av denna PKI-lösning, särskild relaterat till försvarsmässiga behov. Utvecklingen av detta protokoll drivs gemensamt och öppet av olika i sammanhanget centrala aktörer, vilket vi bedömer underlättar uppfyllandet av krav I. (+)

II. Information till rätt plats:

Ett väl implementerat certifikatsystem i en PKI-lösning underlättar just att information är tillgänglig på rätt plats, i rätt tid och för rätt aktör. SPKI/SDSI torde även här ha en extra styrka gentemot PGP. (++)

III. Ansvarshierarki, men ändå icke- hierarkiskt informationsflöde:

Certifikathierarkin i en PKI-lösning bedömer vi kan anpassas väl till en befäls- och ansvarshierarki. SPKI/SDSI och X.509 är mera lämpad för detta än den mera ad hoc-mässiga certifikatdistribution som PGP-system erbjuder. (++)

Delkravet om informationsutbyte oberoende av nivågränser kan dock komma i konflikt med anpassningen till en befäls- och ansvars hierarki. En avvägning behöver göras mellan dessa önskemål. (-)

IV. Informationen nära ägaren:

PKI-lösningar förhindrar inte att information befinner sig nära ägaren, samtidigt som certifikatsystemet kan hantera distribution i den omfattning som är lämplig. (+)

V. Små delsystem:

Gruppbegreppet, vilket är en grundläggande del av SPKI/SDSI, utgör en tydlig fördel för att realisera avknoppning av delar system, insatsstyrkor och autonoma delsystem. (+)

Den globala katalog som åtminstone i ursprungsdefinitionen är basen för X.509, bedömer vi däremot inte underlättar realisering av detta krav. (-)

VI. System av system:

Även för system av system bedömer vi att PKI, och särskild SPKI/SDSI med sin uppbyggnad kring definierade grupper, bör kunna vara ett intressant alternativ. (+)

Sammanfattningsvis utpekar sig *SPKI/SDSI* som ett bland olika PKI-lösningar intressant alternativ för en närmare bedömning. För ett nätverksbaserat försvar bedöms X.509 innebära system som i vissa avseenden blir svårare att implementera och förvalta. PGP är å andra sidan för öppen i sin struktur till att passa militära tillämpningar. Andra alternativ kan existera som vi för närvarande inte har kunskap om och därmed inte har gjort någon bedömning av. Kommersiella programvaror, som till exempel VeriSigns PKI-lösningar, har vi inte bedömt.

5.3 Identitetsbaserad autentisering

I autentiseringsprocessen ingår ett steg som innebär att verifieringsnyckeln på ett säkert och icke-manipulerbart sätt skall kopplas ihop med rätt identitet. Den som skall verifiera att en uppgiven identitet är äkta måste ju vara säker på att han inte blivit lurad att använda fel verifieringsnyckel.

Denna hopkoppling mellan identiter och verifieringsnycklar kan göras på olika sätt. Vid biljettbaserade metoder, jämför Kerberos ovan, finns hopkopplingarna i form av register i betrodda servrar som man kommunicerar med på ett säkert sätt. Vid mer distribuerade metoder, jämför PKI ovan, finns hopkopplingarna beskrivna i kedjor av certifikat som är digitalt signerade av betrodda instanser. Nackdelar med dessa metoder är bl.a. att det är sårbart att förlita sig på kommunikation med server och att det är resurskrävande att hantera certifikatkedjor.

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 algoritmer 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. Nedan beskrivs två stycken, men det finns fler.

Verifieringsnyckeln är alltså den av alla kända identiteten. Det steg av övervakning som måste ske i födelseprocessen (fråga (d) i inledningen av detta kapitel) kan därför 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 kan redan nu påpekas att en väsentlig nackdel med de nedan beskrivna metoderna är att alla aktörer måste administreras av samma administratör. Därmed kan det bli svårt att koppla ihop delsystem och det går alltså inte lätt att skala upp till stora system. FOI har tidigare tagit fram en metod [BEN01] med egenskapen att administrationen kan dele-

geras till flera administratörer. Den beskrivs dock inte här, eftersom det visat sig att metoden inte kan göras tillräckligt säker.

Guillou-Quisquater

Den som först påvisade möjligheten med identitetsbaserad autentisering var Shamir [SHA85]. Shamir är en av upphovsmännen till RSA-kryptering, den i särklass vanligaste metoden för öppen-nyckel kryptering. Det är också i princip RSA-algoritmerna som är grunden i Shamirs nydanande metod. Den har sedan följts av ett antal metoder. De kan sägas vara varianter av samma grundidé, se kapitlet om "Identification Schemes" i [SCH96c]. I [SCH96c] påpekas bland annat att ett av de svåra problemen med identitetsbaserade metoder är att visa att inte flera aktörer i samarbete kan beräkna någon annans hemliga autentiseringsnyckel (vilket är anledningen till att FOI-metoden misslyckades). Detta beror på att nycklarna inte kan väljas slumpmässigt utan måste följa något visst mönster. De måste ju på ett eller annat sätt vara kopplade till identiteterna.

Som ett exempel på metoder med öppet känd verifieringsnyckel har valts Guillou-Quisquater [SCH96C, GUI89]. Den beskrivs kortfattat nedan, utan försök till matematisk stringens.

En betrodd administratör (BA) är den ende som känner primtalsfaktorerna p och q . Han väljer ett tal s och beräknar $n=p*q$. Talen s och n görs allmänt kända för alla aktörer. I alla beräkningar nedan görs beräkningarna modulo n , men för enkelhets skull skrivs inte $(\text{mod } n)$ ut. Den aktör som vill autentisera sig kallas nedan för A , och den som vill verifiera A 's identitet kallas V .

Födelseprocessen tillgår så att BA löser ut talet h ur $A * h^s = 1$. Detta kan ingen annan göra än den som känner primtalsfaktorerna i n , dvs. BA . A tilldelas h , som är hans hemliga autentiseringsnyckel. Hädanefter behöver inte A ha kontakt med BA .

- A vill autentisera sig för V . Han tar kontakt med V och uppger sig heta A .
- V väljer ett slumptal v , där $v < s$, och skickar v till A .
- A väljer ett slumptal a , där $a < n$. Han beräknar $c = a^s$ (s är allmänt känt). Med hjälp av sin hemlighet h och det mottagna v beräknar han också $d = a * h^v$. Till V skickar han de två talen c och d .
- V verifierar A genom att kontrollera att $c = d^s * A^v$ (s är allmänt känt och v har V själv valt).
(Det blir likhet vid verifieringen, eftersom $d^s * A^v$
 $= a^s * h^{sv} * A^v = a^s (A * h^s)^v = a^s * 1 = c$)

Detta är alltså ett exempel på att en part kan kontrollera att motpartens identitet är I , enbart med kännedom om talet I . Men det bygger på att en BA skapar de hemliga autentiseringsnycklarna (eller i och för sig flera BA som alla vet samma hemlighet, nämligen p och q). En annan observation är att beräkningarna ovan är relativt tunga.

Leighton-Micali

Leighton-Micali [LEI94] är ett exempel på system med symmetriska nycklar, dvs. autentiserings- och verifieringsnyckeln är samma. Eftersom en aktör måste hålla sin autentiseringsnyckel hemlig, måste ju då också verifieringsnyckeln hållas hemlig. Denna kan då inte vara identisk med den allmänt kända identiteten I , vilket var fallet i metoden Guillou-Quisquater. I stället beräknas nyckeln med hjälp av en envägsfunktion utgående från dels identiteten I , dels en hemlighet som är unik för varje aktör.

Egentligen är därför Leighton-Micali ett (patenterat) system för (öppen) nyckel-distribution. Detta innebär att enbart de två parterna, med identiteterna A och V , var för sig, och utan att utväxla hemliga data, kan konstruera samma nyckel K_{AV} . Denna använder A för att styrka sin identitet gentemot V , som alltså använder sin beräknade K_{AV} för verifieringen. Själva autentiseringsprotokollet, (fråga (b) i inledningen av detta kapitel), innebär utväxling av slumpstal mellan A och V . Dessa slumpstal ingår i ett meddelande, M_{AV} , som A sänder till V . Med hjälp av sin version av K_{AV} beräknar A en digital signatur utav M_{AV} och sänder också signaturen till V . Denne bildar utifrån det mottagna M_{AV} sin version av den digitala signaturen, och verifierar att de två signaturerna är identiska.

Om det finns n aktörer i systemet, som alla skall kunna autentisera sig mot varandra, måste alltså var och en kunna skapa $n-1$ st olika nycklar för sin autentisering och $n-1$ olika nycklar för verifiering av motpartens identitet. Metoden att konstruera dessa nycklar, ovan kallade K_{AV} , beskrivs inte här i detalj eftersom det kräver en del matematiska begrepp (se [LEI94]). Men det bör påpekas att beräkningarna är enkla. I de olika stegen används en och samma envägsfunktion applicerad på en identitet och en hemlighet. Ett naturligt val av funktion är den standard som kallas HMAC [KRA97, BIH92]. I ett steg används också den enkla operation som brukar betecknas XOR.

Den viktiga födelseprocessen sköts av en enda (eller flera som vet samma hemligheter) betrodd administratör, BA. Denne har valt två grundhemligheter, K och L . Utifrån dessa tilldelar BA varje aktör, t.ex. A , ett nytt par av grundhemligheter, kallade K_A och L_A . Hädanefter behöver i princip inte A ha kontakt med BA. Men för att kunna beräkna den gemensamma nyckeln, K_{AV} ovan, måste A och V använda två värden som finns på platsen rad= A , kolumn= V , i två tabeller som båda måste skapas av BA. Dessa två tabeller kan bara BA konstruera och de måste alltså distribueras i systemet. Dessa tabeller är inte hemliga. Ett sätt att distribuera dem är, att vid födelsen får en aktör de tabeller som gäller för de aktörer som hittills fötts. När två aktörer vill ha kontakt har då den yngste de nödvändiga tabellvärdena, som han kan skicka till den äldre. Men då skulle han kunna maskera sig genom att skicka värden för en annan ung aktör och sedan agera i dennes namn. Detta är anledningen till att det finns två tabeller. De beror av varandra, och på identiteterna, på ett sätt som förhindrar maskering. Metoden blir identitetsbaserad i och med att de enda variabla parametrar som finns i beräkningarna är de två aktörernas identiteter.

5.3.1 Egenskaper i relation till kravlista

De två här beskrivna 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 kravlistan, i avsnitt 3.1, 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 kravlistan.

I. Civil teknik:

Det är fråga om civilt baserad och plattformsoberoende teknik. Däremot är tekniken inte standardiserad eller frekvent implementerad. (-)

II. Information till rätt plats:

Ingen auktoriseringsinformation finns, förutom identiteten. (Det finns dock möjlighet att koda in viss information). (0)

III. Ansvarshierarki, men ändå icke-hierarkiskt informationsflöde:

Svårt att hantera hierarkier i och med beroendet av en enda administratör. (--)

IV. Information nära ägaren:

Inget hinder. (+)

V. Små delsystem:

Går ganska lätt att sätta upp i och med att metoderna är självförsörjande och lokala. (++)

VI. System av system:

Mycket svårt att koppla ihop olika system. (--)

6 Metodernas för- och nackdelar i relation till olika försvarsuppgifter

I kapitel 3 diskuterades olika försvarsuppgifter som det framtida nätverksbaserade systemet skall understödja. De olika uppgifterna medför ett antal krav på det distribuerade informationssystemet. Kapitel 3 avslutades med en lista, I-VI, av sådana krav. Listan gör inte anspråk på att vara fullständig, men den kan ändå tjäna som stöd för en jämförelse mellan olika klasser av autentiseringsmetoder. Listan upprepas nedan, kompletterad med kommentarer hur de tre tidigare beskrivna klasserna av metoder förhåller sig till respektive krav.

Det bör, rörande bedömningarna vilka är sammanfattade i form av markeringar med plus och minus, påpekas att dessa bedömningar har en begränsad precision. Man kan till exempel inte rakt av räkna antal plus för en metod och ur detta enkelt besluta vilken metod som är bäst. Dessutom har bedömningarna gjorts för en grupp av metoder. Detta kan betyda att en enskild metod inom en klass bedömts som märkbart bättre eller sämre inom gruppen, vilket i sin tur påverkar helhetsbedömningen.

Det bör också påpekas att man svårligen kan realisera ett system baserat på flera av autentiseringssystemen, ty de kan ej samarbeta. Med andra ord, nackdelar hos en metod kan ej uppvägas av motsvarande fördelar hos en annan genom att implementera båda metoderna.

- I. 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 Internet-teknik, som på något sätt måste göras "tillräckligt säker". Flexibilitetskrav gör att man skall välja leverantörs- och plattformsoberoende lösningar.

Bedömning av för- och nackdelar i sammanfattande betyg:

Biljettbaserade: ++ Certifikatbas.: + Identitetsbas.: -

Kommentarer:

Termen "tillräckligt säker" ger ett tolkningsutrymme. I försvarsspecifika sammanhang är ofta kraven hårdare specificerade och högre än i civila sammanhang. Detta påverkar på vilket sätt och i vilken omfattning civila lösningar kan användas. Man kan dock konstatera att utvecklingen går mot ett ökande inslag av civil teknik och civila system inom det militära.

Med avseende på detta krav har Kerberos tydliga fördelar, speciellt genom den erfarenhet man har av lösningen genom ett antal implementeringar. PKI kan dock bli en intressant lösning, dels genom pågående forskningsprojekt och dels genom ett ökande antal kommersiella PKI-lösningar.

Med andra ord torde kommande utveckling medföra att dessa två metoder torde vara lika intressanta i relation till krav I. Identitetsbaserade system bedöms dock relativt krav I vara mindre intressanta.

- II. 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.

Bedömning av för- och nackdelar i sammanfattande betyg:

Biljettbaserade: +/- Certifikatbaserade: ++ Identitetsbaserad: 0

Kommentarer:

Om man väger in hela kravformuleringen med bland annat formuleringen "på rätt plats, i rätt tid och för rätt aktör" torde PKI vara den starkare kandidaten. Det bör dock noteras att man då har utvidgat bedömningen till att omfatta metodernas auktoriseringsförmåga, inte endast autentiseringsförmågan. Frågorna om autentisering och auktorisering är relaterade, och vid val av metod torde det vara svårt att exkludera en av frågorna. Kerberos har också vissa möjligheter till auktoriseringsförmåga men är starkt beroende av att servern är tillgänglig. Identitetsbaserade metoder har ingen auktoriseringsförmåga.

Rörande den sekundära uppgiften att möjliggöra styrning av sensorer och plattformar, så har den endast bedömts relativt autentiseringsförmågan. Själva styrningsförmågan ligger med andra ord utom ramen för denna rapport.

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

Bedömning av för- och nackdelar i sammanfattande betyg:

Biljettbaserade: + Certifikatbaserade: ++ Identitetsbaserad: --

Kommentarer:

Både PKI och Kerberos är intressanta metoder relativt krav III. Eventuellt har PKI en något starkare position. Det bör noteras att informationsutbyte oberoende av nivågränser medför en balansgång. Detta behov är ej enkelt att realisera i ett autentiseringssystem som i grunden baserar sig på och skall avspegla en hierarki.

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

Bedömning av för- och nackdelar i sammanfattande betyg:

Biljettbaserade: - Certifikatbaserade: + Identitetsbaserad: +

Kommentarer:

Med avseende på detta krav har PKI och identitetsbaserade metoder vissa fördelar. Biljettbaserade metoder är serverorienterade. Detta får till följd att de också är mer lämpade när informationen finns i servrar än när informationen finns nära ägaren.

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

Bedömning av för- och nackdelar i sammanfattande betyg:

Biljettbaserade: -- Certifikatbaserade: +/- Identitetsbaserad: ++

Kommentarer:

Med avseende på detta krav framträder identitetsbaserade metoder tillsammans med SPKI/SDSI som de mest intressanta kandidaterna. Övriga bedömda PKI-lösningar samt Kerberos är relativt detta krav mindre intressanta kandidater.

- VI. 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.

Bedömning av för- och nackdelar i sammanfattande betyg:

Biljettbaserade: ++ Certifikatbaserade: + Identitetsbaserad: --

Kommentarer:

Med avseende på detta krav har Kerberos fördelar. PKI bedöms vara intressant och då särskild SPKI/SDSI. Det är mycket svårt att koppla ihop system som är baserade på identitetsbaserade metoder.

6.1 Slutsatser

Tabell över presenterade metoder jämfört med krav I-VI.

	Biljett-baserade	Certifikat-baserade	Identitets-baserade
I. Civil teknik	++	+	-
II. Informationstillgänglighet	+/-	++	0
III. Hierarki	+	++	--
IV. Informationsägande	-	+	+
V. Delsystem	--	+/-	++
VI. System av system	++	+	--

Tabell 6.1 Värdering av autentiseringsklasser

Bedömningarna med plus och minus ger en fingervisning av vad den övergripande klassen kan klara av, men variationer finns inom klassen. Exempelvis ger certifikatklassen intryck av att vara den mest anpassningsbara modellen, men då skall man ha i åtanke att det finns flera olika implementeringar att välja på och de ger olika resultat. Tabellen är en grov generalisering.

Utifrån detta material bedömer vi att biljettmetoder som Kerberos och certifikatmetoder (PKI) är de kandidater som är mest intressanta för att stödja realiseringen av det nätverksbaserade försvaret.

7 Referenser

- [ALL99] C. Allen, T. Dierks, "The TLS Protocol Version 1.0", RFC 2246, januari 1999, <ftp://ftp.isi.edu/in-notes/rfc2246.txt>, besökt 6 December 2001.
- [BEN00] A. Bengtsson, J. Hallberg, D. Lindahl, U. Lindbergh, M. Persson, "Tillämpad nätverkssäkerhet", Underlagsrapport FOA-R--00-01472-505--SE, FörsvaretsForskningsanstalt, 2000.
- [BEN01] A. Bengtsson, "An attempt to get identity-based authentication and certification in a hierarchic tree", Scientific report FOI-R--0114--SE, mars 2001, Stockholm, Totalförsvarets Forskningsinstitut.
- [BIH92] E. Biham, "New Types of Cryptanalytic Attacks Using Relating Keys", Technical Report #753, Computer Science Department, Technion-Isreal Institute of Technology, September 1992.
- [CLA99] D. Clarke, J-E. Elie, C. Ellison, M. Fredette, A. Morcos, R. L. Rivest, "Certificate Chain Discovery in SPKI/SDSI", 1999. <http://theory.lcs.mit.edu/~cis/sdsi.html>, besökt 6 December 2001.
- [ELI98] J-E. Elie, "Certificate Discovery Using SPKI/SDSI 2.0 Certificates", examensarbete, 1998. <http://theory.lcs.mit.edu/~cis/sdsi.html>, besökt 6 December 2001.
- [FJÄ00] P-O. Fjällström, G. Neider, M. Persson, T. Risch, P. Svensson, "Arkitekturprinciper för informationsöverlägsenhet i framtidens ledningssystem", Användarrapport FOA-R--00-01435-505--SE, Försvarets Forskningsanstalt , februari 2000.
- [FM99] Försvarmaktsidé 2020, Rapporterna 1-5, Årsrapporter från perspektivplaneringen Stockholm, Försvarets bok- och blankettförråd.
- [FM00] Försvarsplan 2000
- [FM01] Försvarmaktens grundsyn ledning 2001, Stockholm, FM 2001, 28 s.
- [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 Constituent autant d'Accréditations", Proceedings of SECURICOM '89, sid 149-158, 1999.

- [HAR92] D. Hartman, "Unclogging Distributed Computing", IEEE Spectrum, 29(5), sid 36-39, maj 1992.
- [ITS] ITS, Informationstekniska standardiseringen I Sverige, "Terminologi för Informationssäkerhet", Rapport ITS 6, mars 1994.
- [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.
- [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.
- [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.
- [OPEN] <http://www.openp2p.com>.
- [PAR01] M. Parameswaran, A. Susarla, A. B. Whinston, "P2P Networking: An Information-Sharing Alternative", IEEE Computer, juli 2001.
- [PURL] <http://www.purl.org>.
- [RIV98] R.L. 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 571-572.

- [SCH96c] B. Schneier, Applied Cryptography: Protocols Algorithms and Source Code in C, Sec. Ed., John Wiley & Sons, 1996, sid 508-510.
- [SDSI] <http://theory.lcs.mit.edu/~cis/sdsi.html>.
- [SHA85] A. Shamir, "Identity-Based Cryptosystems and Signature Schemes", Advances in Cryptology: Proceedings of CRYPTO 84, Springer-Verlag, sid 47-53, 1985.
- [SIG98] Säkerhetsarkitekturer, Dataföreningen i Sverige, SIG Security, 1998, sid 60-63.
- [SPKI] <http://world.std.com/~cme/html/spki.html>.
- [TST01] Handbok Totalförsvarets Signalskyddstjänst (Grundläggande regler för signalskyddstjänsten – H TST Grunder 2001), 12 839:62574.