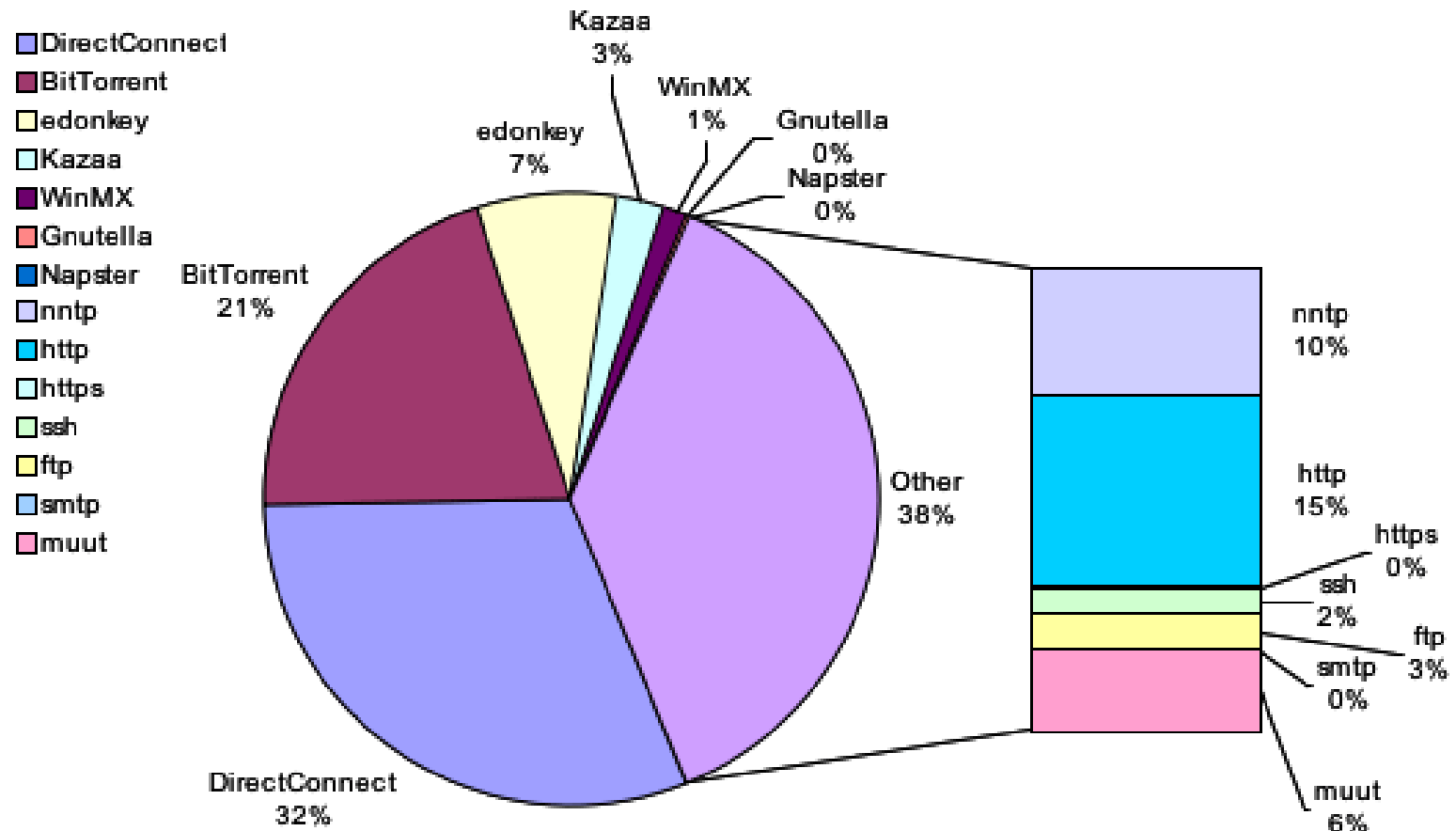


Operativsystem

Säkerhet
(kap 9 i boken)

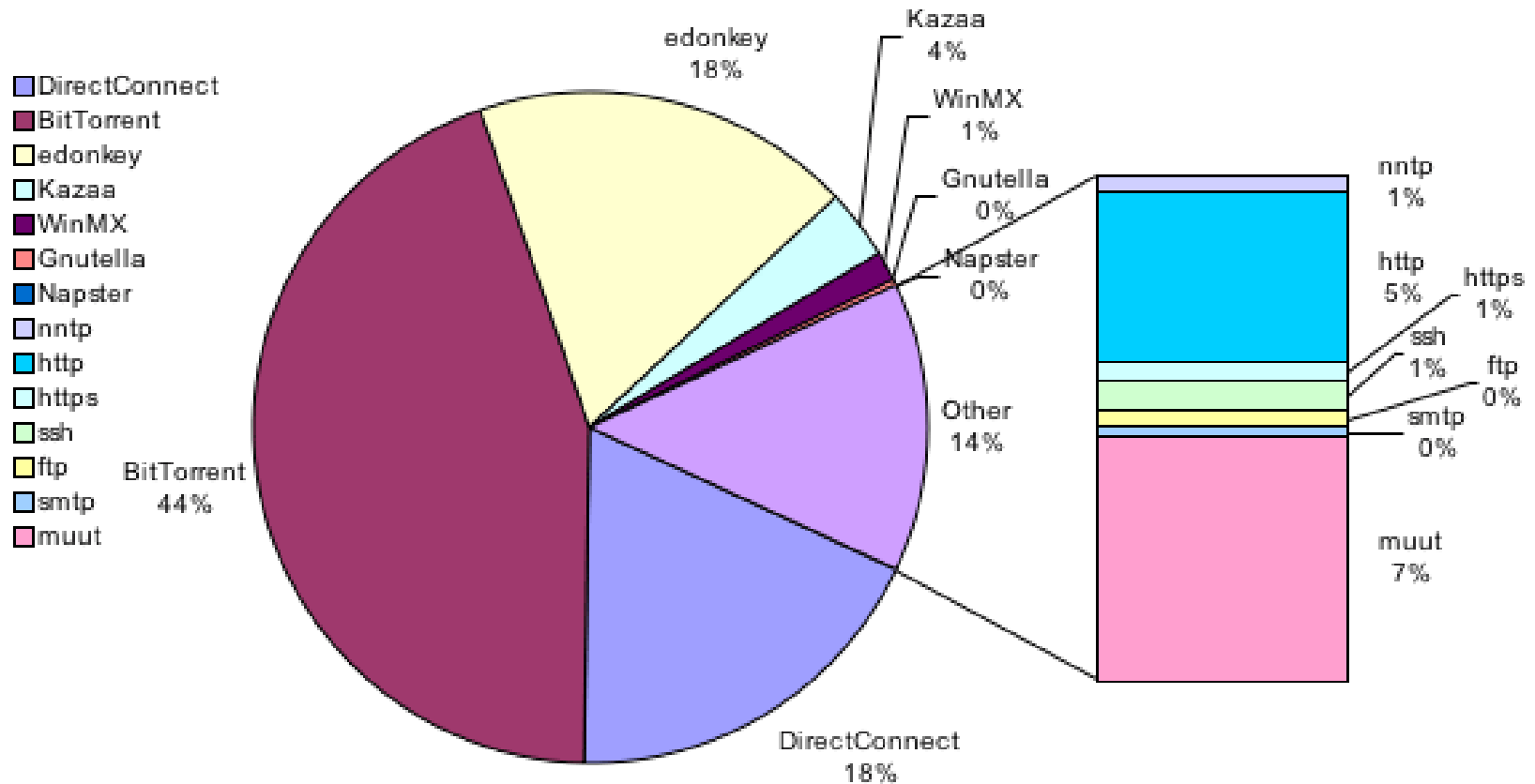
(Nätanvändning TY)

Yliopiston liikennejakauma 12.10.2004 - 19.10.2004, sisääntuleva liikenne



(Nätanvändning TY)

Yliopiston liikennejakauma 12.10.2004 - 19.10.2004, ulospäinsuuntautuva liikenne



Vad är säkerhet?

Säkerhet i datasystem kan ses från två aspekter

- 1. Att information/system finns tillgänglig då man behöver den**
- 2. Att ej obehöriga kommer åt informationen/system**

Vilken är den viktigare???

(Då man talar om säkerhet är det nästan alltid det senare det talas om, på bekostnad av det tidigare)

Information och åtkomst

- Information är värdefull endast när någon har möjlighet att komma åt den (man måste säkerställa åtkomsten)
- Det att obehöriga kommer åt information kan innebära
 - Ekonomiska förluster (minskade inkomster)
 - Den personliga integriteten skadas (t.ex. databaser med personuppgifter)

Säkerställa åtkomst

- RAID-filsystem
- Backups (vad om huset brinner upp??)
- Dubblerade datorsystem
- Uninterruptible Power Supplies (UPS)
- Dubblerade nätverksförbindelser
 - Internet egentligen designat som ett säkert nätverk
- Multipla sätt att komma åt data (web, filsystem, databaser)
- Dokumentering av system

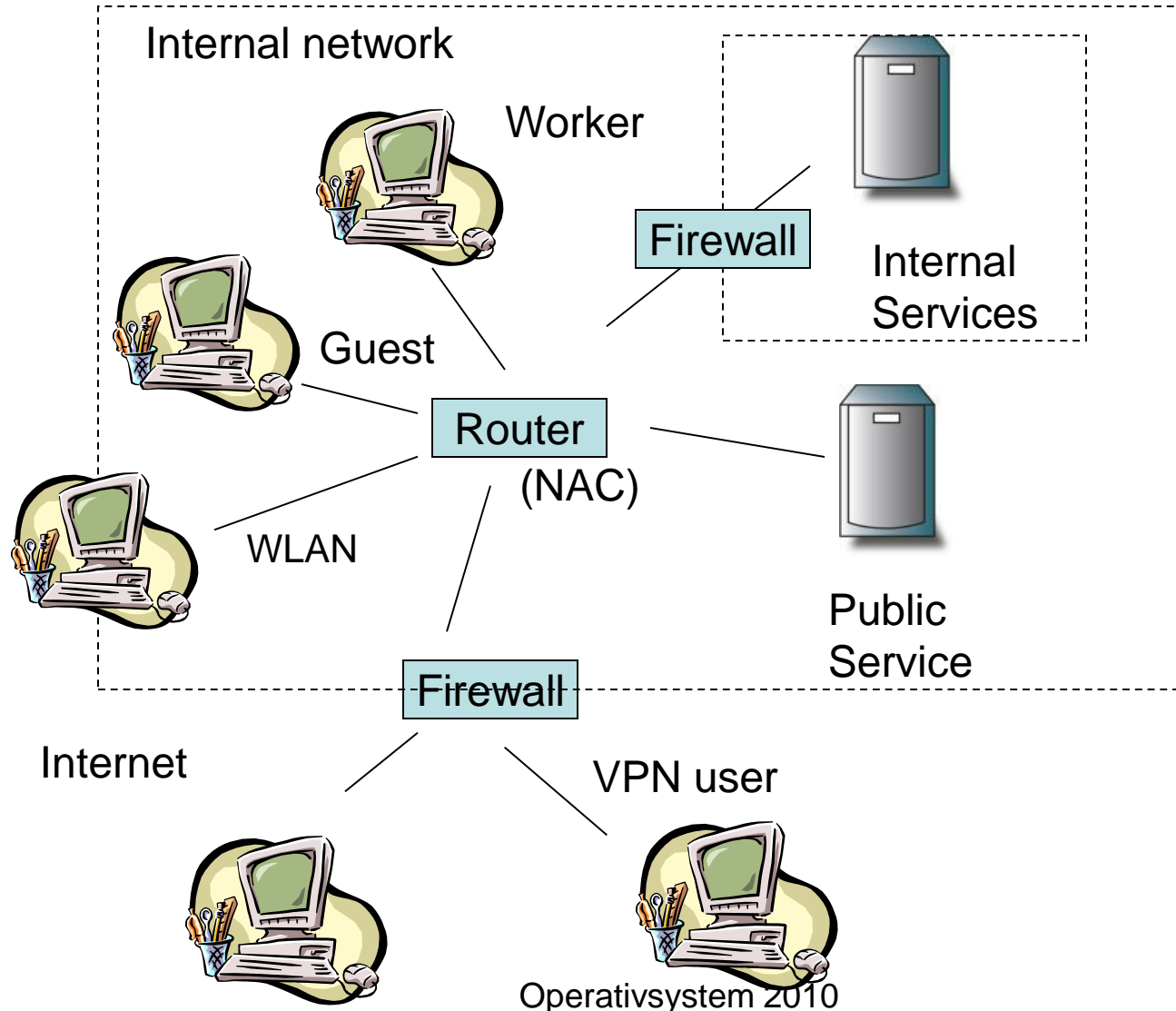
Skydd av system

- Dataintrång i system kan föranleda
 - Åtkomsten av data blir förhindrad (t.ex. p.g.a. att systemet är nere, eller att data blir förstörd)
 - Att obehöriga kommer åt data som de inte har rätt till

Typer av system vi vill skydda

- Personliga datasystem
 - Hemdator / jobbdator (bärbar??)
 - Användningen av datorn / filerna
- Servicepunkter
 - Servrar / databassystem
 - Epost-system
- Nätverksresurser

Metoder för att skydda



Metoder för att skydda

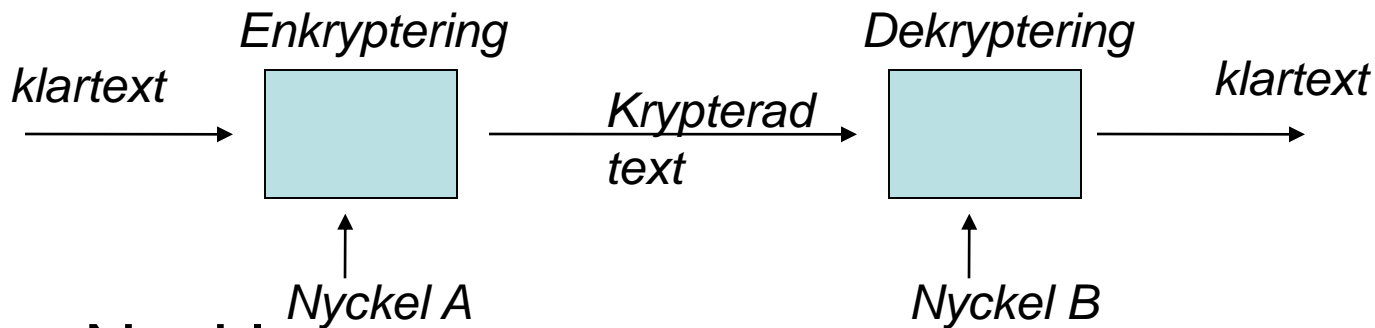
- Fysiska krav (krav på var användaren måste befinna sig)
- Logiska krav (vilket nätverk användaren använder)
- Krav på nätverket (t.ex. enkryptering)
- Autenticering
- Auktorisering
- Krav på den maskin som man använder för att accessera resurser (virus-skydd)

Dataintrång

- Läsa över axeln, dator med öppna meddelanden
- Interna försök att se hur väl system är skyddade
- Externa försök, försök att "överta" maskiner
- Intrång med ekonomiska motiv
- Direkt spionage

Kryptografi

- Grunderna för kryptografi



- Nycklar:
 - Symmetriska: Nyckel A = Nyckel B
 - Publika: Nyckel A \neq Nyckel B
- Själva krypteringsmetoden känd, nycklarna okända

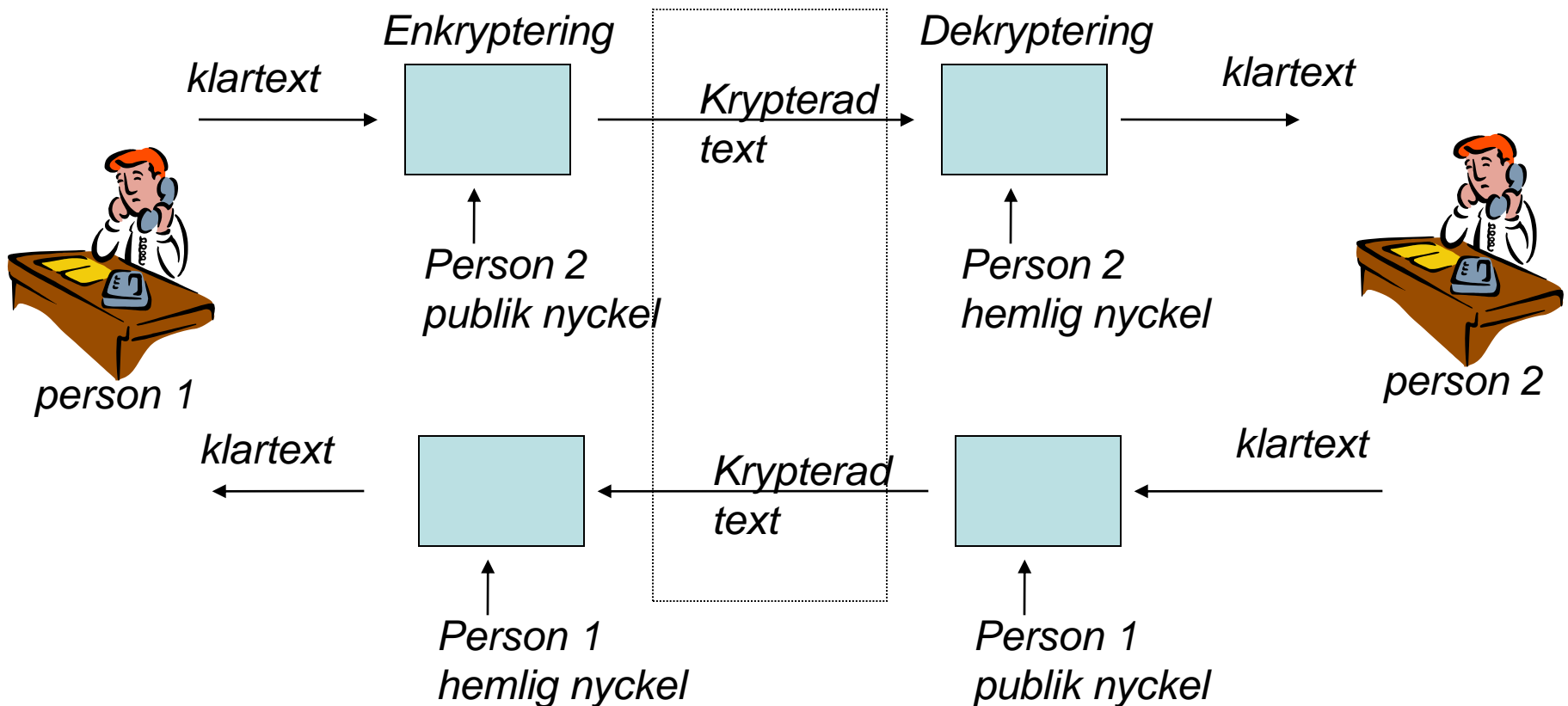
Symmetriska nycklar

- Två personer vill kommunicera hemligt
- de börjar med att utbyta nycklar
- sedan kan de använda nyckeln för att skicka meddelande över t.ex. en publik datalänk
- Nyckellängd: t.ex. 1024 bit ger en tillräkligt stor nyckelrymd
- Problem: Hur överföra nyckeln så att obehöriga ej får den?

Publika nycklar

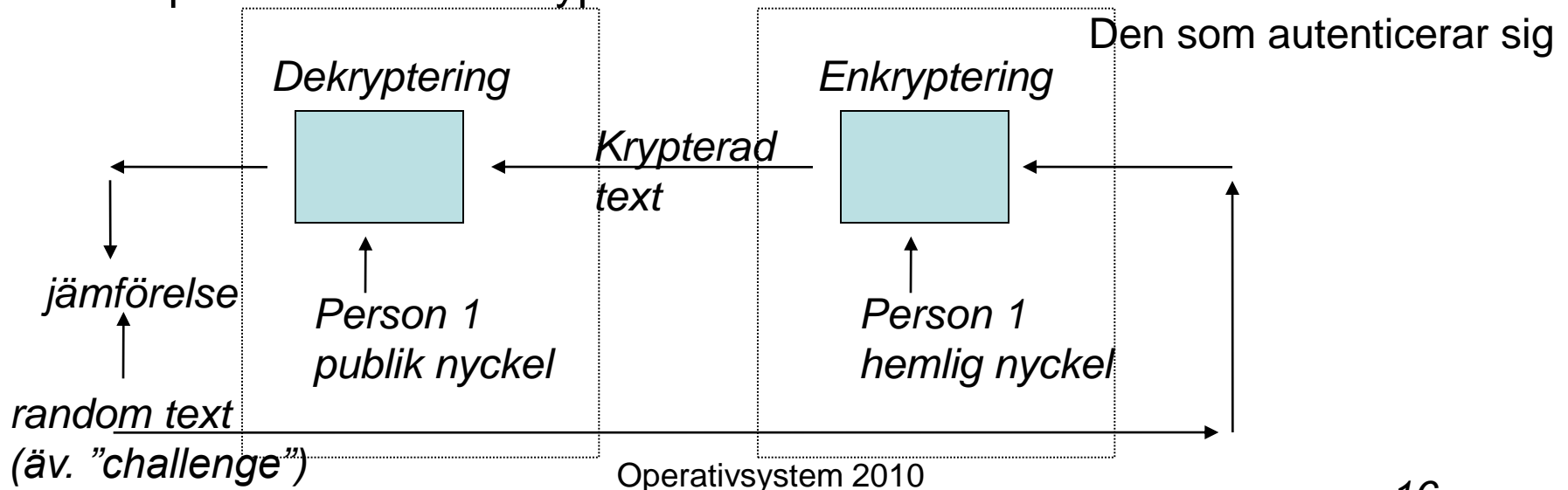
- Skapas ett nyckelpar: en publik nyckel A samt en hemligt nyckel B
- Detta sker vanligen på personbasis, en person skapar detta nyckelpar: den publika publiceras, den hemliga håller personen i gott förvar
- Detta system kan nu användas för
 - Skicka krypterade meddelanden
 - Autentikering
 - Digital underskrift

Skicka krypterade meddelanden



Autentikering

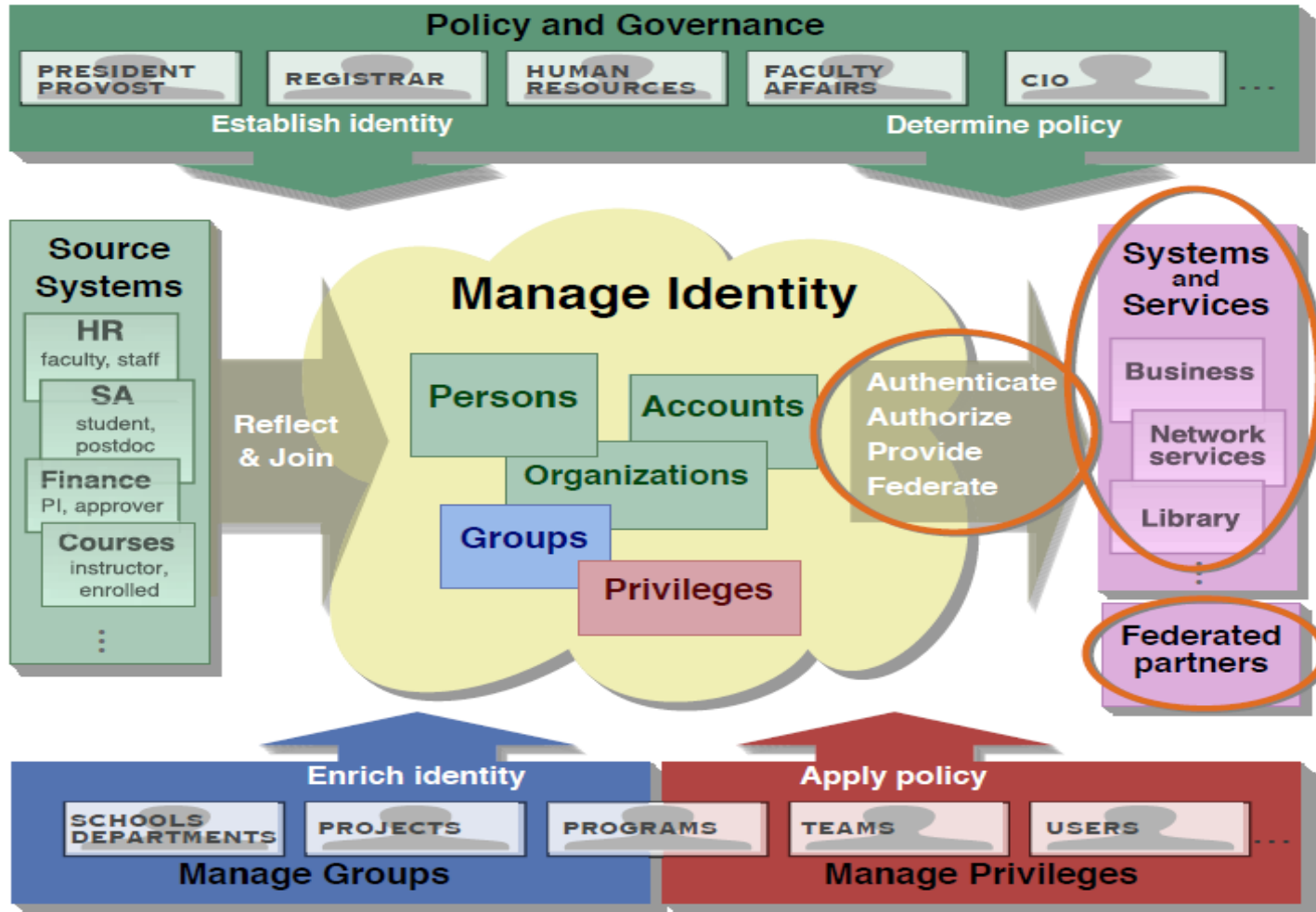
- Bevisa vem du är!!
- Hur? Genom att känna till något som endast du har tillgång till
 - Lösenord: Egentligen är det idiotisk att bevisa att man känner till lösenordet genom att mata in det i systemet
 - Istället: Bevisar att personen har den hemliga nyckeln genom att personen kan dekryptera ett meddelande



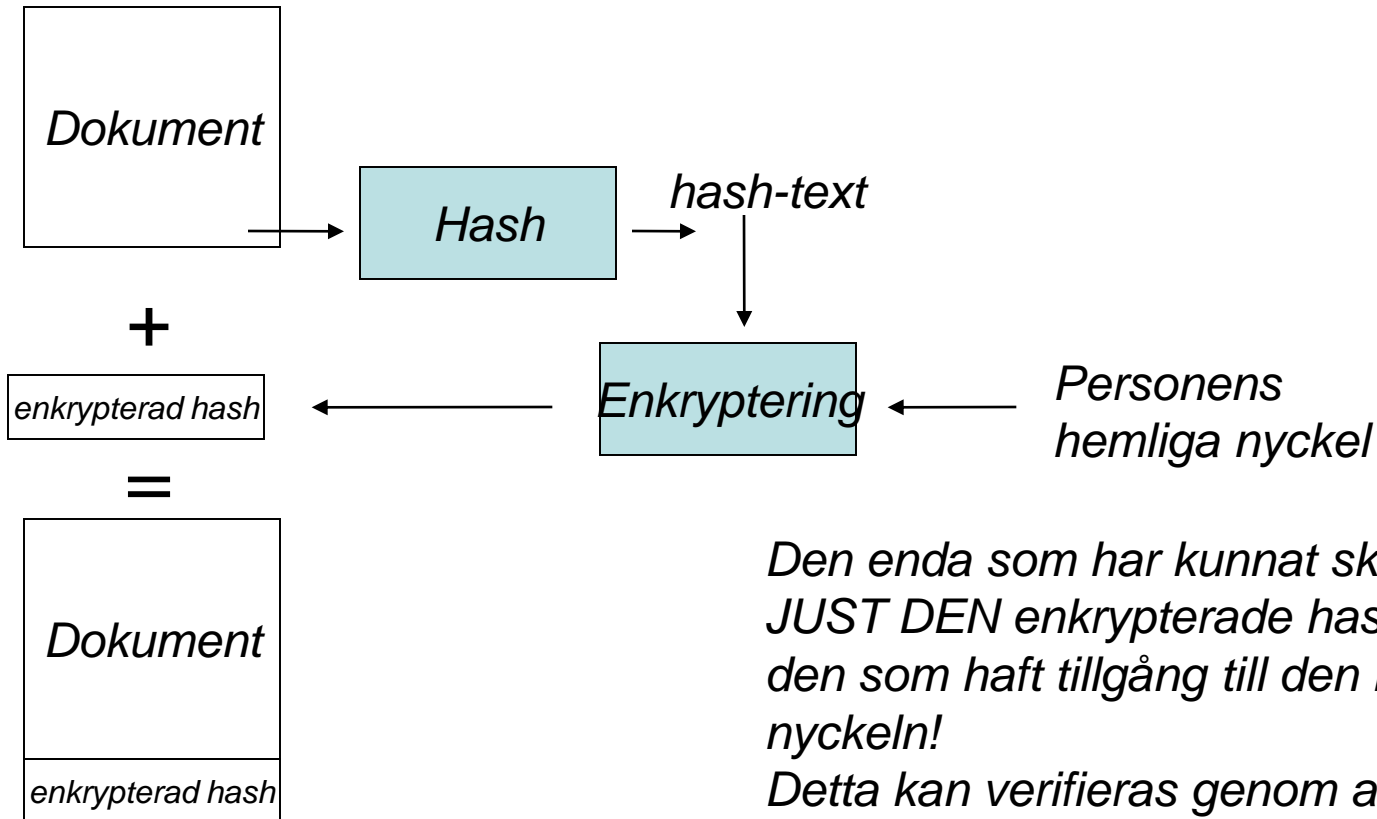
Auktorisering

- "OKEJ, vi vet vem du är, men nu måste vi avgöra vad du får göra i vårt system"
 - ex.
 - Filpermissioner
 - Access control lists
 - Databasrättigheter
- Vanligt problem: Kan auktoriseringssystemet lita på autenticeringssystemet (hur många gånger måste man "logga in" per dag???)
 - Jfr. Single-login-system
 - Finska universitet: Shibboleth

Shibboleth



Digital underskrift



*digitalt underskrivet
meddelande*

*Den enda som har kunnat skapa
JUST DEN enkrypterade hashen är
den som haft tillgång till den hemliga
nyckeln!*

*Detta kan verifieras genom att med
personens publika nyckel decryptera
hashen; är den rätt?*

Kombination av publik och symmetrisk nyckel

- Enkryptering/dekryptering med publik/hemlig nyckel är relativt långsamt
- För datakommunikation används ofta symmetrisk enkryptering
- Vid datakommunikation:
 1. Autenticering, pratar vi med rätt person?
 2. Hitta på en random symmetrisk nyckel, överför denna med publik/hemlig nyckel-systemet
 3. Använd den symmetriska nyckel för fortsatt kommunikation (snabb!!)
 4. Byt den symmetriska nyckel med jämna mellanrum!

Exempel: https (dvs Secure Sockets layer (SSL)-baserad kommunikation)

FINEID



- I Finland används digitalt personkort
- Använder sig av publik/hemlig nyckel
- Den hemliga nyckeln finns inprogrammerad i kortet
 - Kortet kan enkryptera meddelande med hjälp av nyckeln; själva nyckeln kan ej läsas från kortet
 - Möjligheten att enkryptera är skyddad med en PIN-kod
- Varje person associeras med ett digitalt certifikat – ett dokument som är digitalt underskrivet av befolkningsregistercentralen – äktheten på certifikat kan lätt granskas
- Försvunna kort? Befolkningsregistercentralen upprätthåller en "Certificate Revocation List" – förrän ett certifikat godkänns checkar man mot denna lista

Lösenord

- Används för autentikering
- Problem
 - Mänskor använder ofta lösenord som är lätta att komma ihåg
 - Namn på hund, något trevligt ord, fruns systems namn etc.etc.
 - De flesta system kräver att du ger ditt lösenord (fast det endast är fråga om att kontrollera att du kan ditt lösenord)

Lösenord UNIX

- Lösenorden sparas i en fil /etc/passwd i enkrypterad form

```
superuser:SxGUT0Jktx60c:12121:0:99999:7:::
```

- Enkrypterat password "SxGUT0Jktx60C"
- Encryptering sker med funktionen
 - `char *crypt(const char *key, const char *salt)`
- Från de första 8 tecknena i lösenorden används 7 bitar - > 56 bitars lösenord som enkrypteras
- Dessutom används i unix ett "salt", två första tecknena i det enkrypterade lösenordet modifierar enkrypteringalgoritmen
- Genom att jämföra returvärdet från `crypt()` med det sparat i /etc/passwd kan man autentikera användare

Hur länge tar det att bryta in sig i UNIX

- 56 bitars nyckel
- Maskinen på bilden:
56 timmar (1998)
(EFF DES Cracker)
- 1999 22 timmar
- I praktiken enklare p.g.a
av att människor använder
logiska ord som lösenord
- FPGA-version
COPACOBANA 6,4 dagar
(2007)



Shadow passwords

- Problem: /etc/passwd är läsbar av alla->enkelt att försöka hitta giltiga lösenord
- Kan vi gömma /etc/passwd: Nej, många program behöver information som finns i /etc/passwd
- Lösning 2: Flytta encrypterade lösenordet till en skild fil /etc/shadow, som endast root-användaren får läsa

Engångslösenord

- T.ex. Nordea
- Princip: Du bevisar att du har tillgång till den kodtabell banken skickat genom att kunna nästa nummer i tabellen

Challenge respons

- Servern erbjuder en fråga
 - Vad heter din systems förra man?
 - Vilken färg är fantomens vita häst?
- Genom att servern har en lista med frågor / svar kan en användare autentiseras
- Notera: Publik / hemlig nyckel används även på detta sätt
 - Challenge: Enryptera detta meddelande med din hemliga nyckel
 - Om svaret är rätt är användaren autentikerad

Biometrisk autentikering

- Fingeravtryck
- Röstavtryck
- DNA
- Namnteckning
- Näthinna
- (Luktanalys...)

Metoder för "attacker" - insiders

- Trojanska hästar
 - Se ut att vara något annat
 - t.ex. modifiera sökstigen för program
- Login-avlyssning
 - Installera ett pseudo-login-program, får lätt reda på användarnamn / lösenord
- Trap door / login bombs
- Buffer overflow - utnyttja

Virus

- Utnyttjar svagheter i målsystemet för att placera sig där
 - oskyddad bootsector
 - program som är slarvigt programmerade
 - system som inte är konfigurerade på rätt sätt
 - scripting-språk
- Utnyttjar målsystemets resurser för att placera sig på nya målsystem
 - nätverksförbindelser
 - adressböcker

Anti-Virus

- Sök efter bit-kombinationer som typisk representerar Virus (jfr DNA...)
 - Sluga Virus modifierar sig själv så att funktionen hålls intakt, men koden (dvs bit-mönstret) ändrar
- Beräkna checksuma för givna filer, om denna ändrar: Infektion?
- Övervaka beteende – använder något program plötsligt nätet el. dyl.?

Sandlådor

- Ge program endast en begränsad mängd operationer de är tillåtna att utföra
- Begränsa användning av t.ex.
 - filsystem
 - nätverk
 - I/O-enheter
 - minne
- Java använder sandlåde-modellen
- Men i princip: ett väl konfigurerat system använder också sandlåde-modellen