Forelesning 8

Kerberos

Problemstilling

- Et vanlig datanettverk er usikkert - passord som sendes over nettet kan avlyttes av andre og senere misbrukes
- Hvordan kan en bruker bevise sin identitet overfor en datamaskin over et usikkert nettverk, uten å sende et passord i klartekst?
- Hvordan kan en bruker være sikker på at kontakt er opprettet med den riktige datamaskinen?

Kerberos

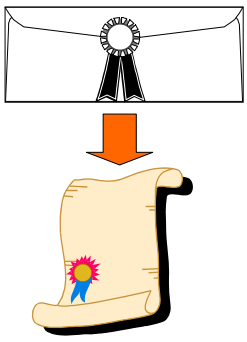
- Utviklet ved MIT på 80-tallet
- Hver bruker autentiserer seg for hver tjeneste som skal benyttes
- Autentisering er gjensidig
- Basert på symmetrisk kryptering
- Med Win2k ble Kerberos introdusert som standard autentisering i Windows Domain

Hva er en billett?

- En vanlig billett er et bevis for noe - f.eks. at du har betalt for å få lov til å reise en tur med bussen
- En vanlig billett blir samlet inn/stemplet/ ødelagt slik at den ikke kan brukes flere ganger
- En vanlig billett er vanskelig å kopiere

Et leidebrev

- Konvolutten er lukket med kongens segl
- Brevet kan ikke åpnes uten å ødelegge seglet
- Hvis brevet inne-holder kontroll-informasjon, kan mottaker verifisere at bæreren ikke har stjålet brevet

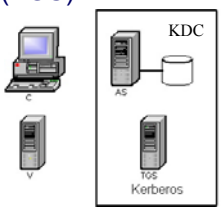


Hva er en Kerberos ticket?

- En ticket er en slags elektronisk billett
- Elektroniske data er lette å kopiere - også for "observatører"
- En Kerberos ticket baserer seg derfor mer på "leidebrevprinsippet"
- I stedet for et segl, "låses" innholdet inn med kryptering
- "Kontrollspørsmålet" er en sesjons-nøkkel som ligger lagret inne i ticket'en

Aktører i Kerberos-protokollen

- Authentication Server (AS)
  - Har tilgang på alle passord
  - Deler en unik nøkkel med alle TGS
- Ticket Granting Server (TGS)
  - TGS utsteder tickets til autentiserte brukere
- ➢AS og TGS kalles KDC
- Client (C)
- Server (V)



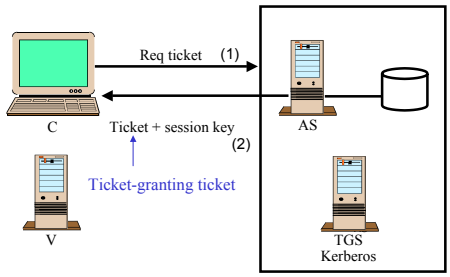
Nøkler i Kerberos

- Langtidsnøkkel (Long-term key), basert på hash av brukerens passord (brukes bare ved pålogging)
- Sesjonsnøkkel (Session key), utstedes av KDC
  - Hver ticket inneholder en (unik) sesjonsnøkkel som serveren bruker til å dekryptere informasjon fra klienter med
  - Totalt mange forskjellige sesjonsnøkler!

Kerberos steg for steg

- Klient C kontakter først autentiserings-server AS for å få TGT
- Kontakter deretter TGS for å få ticket for gitt tjeneste (eller server, V)
- Kontakter til slutt V for å få tjenestesesjon

Hver gang bruker logger på



Hver gang bruker logger på (forts)

(1) Fra C til AS:
PA || Options || ID_C || $Realm_C$ || ID_{TGS} || Times || $Nonce_1$

(2) Fra AS til C:
$Realm_C$ || ID_C || $Ticket_{TGS}$|| $E_{Kc}[K_{C,TGS}$ || Times || $Nonce_1$ || $Realm_{TGS}$ || $ID_{TGS}]$

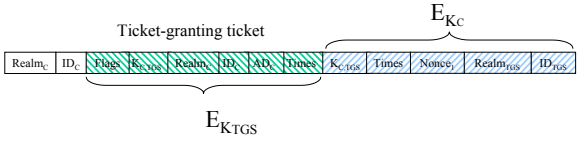
$Ticket_{TGS}= E_{KTGS}[Flags || K_{C,TGS} || Realm_C || ID_C || AD_C || Times]$

Times:
- from : ønsket start-tid
- till : ønsket utløpstid
- rtime : ønsket tidsfrist for fornyelse

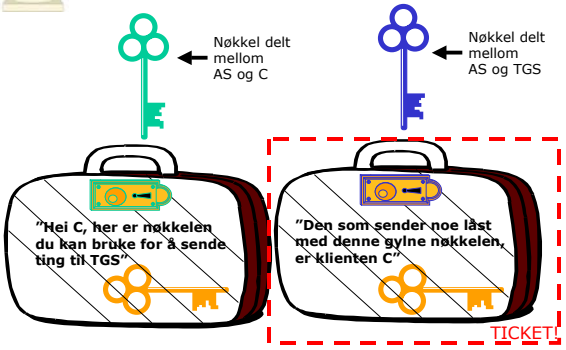
PA: Pre-Authentication data (optional)

Pålogging forts. forts.

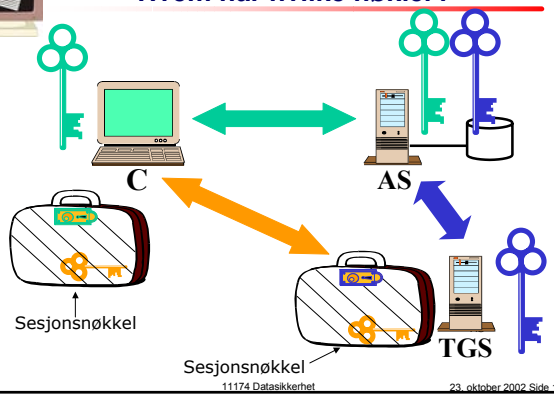
Pakkeformat for svar fra AS til C



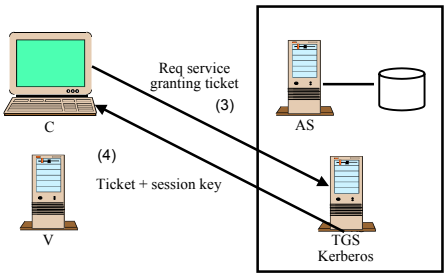
Hva sender AS til klienten?



Hvem har hvilke nøkler?



Hver gang bruker aksesserer en ny tjeneste



Bruker aksesserer en ny tjeneste (forts.)

Ticket-granting ticket

(3) Fra C til TGS:

Options || ID_C || Times || $Nonce_2$ || $Ticket_{TGS}$ || $Authenticator1_C$

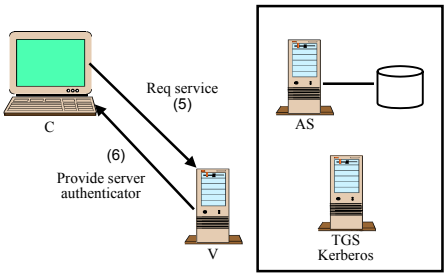
(4) Fra TGS til C:

$Realm_C$ || ID_C || $Ticket_V$ ||
$E_{K_{C,TGS}}[K_{C,V}$ || Times || $Nonce_2$ || $Realm_V$ || $ID_V]$

$Ticket_V = E_{K_V}[$ Flags || $K_{C,V}$ || $Realm_C$ || ID_C || AD_C || Times]

$Authenticator1_C = E_{K_{C,TGS}}[ID_C$ || $Realm_C$ || $TS_1]$

For hver tjenestesesjon



For hver tjenestesesjon

(5) Fra C til V:

Options || $Ticket_V$ || $Authenticator2_C$

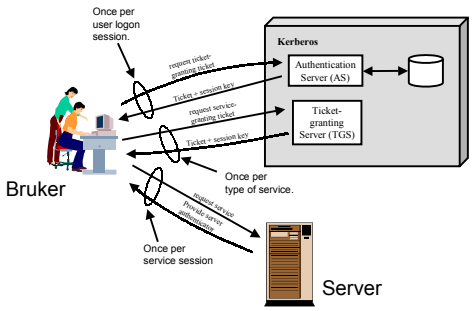
(6) Fra V til C:

$E_{K_{C,V}}[TS_2$ || $Subkey_V$ || Seq#]
(for gjensidig autentisering)

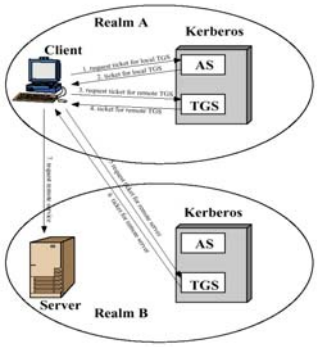
$Authenticator2_C = E_{K_{C,V}}[ID_C$ || $Realm_C$ || TS_2 || $Subkey_C$ || Seq#]

$Subkey_V$ kan utelates, men hvis den finnes, tar den presedens over $Subkey_C$

Kerberos oversikt



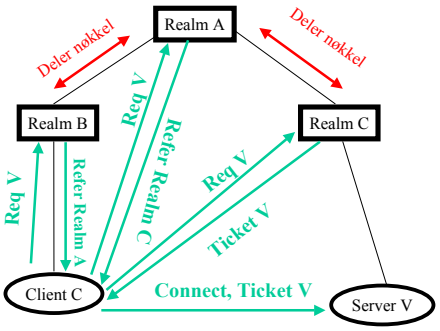
Cross-realm autentisering



Klienten i Realm A ønsker å bruke en tjeneste i Realm B

TGS i Realm A deler en nøkkel med TGS i Realm B

Cross-realm hierarki



Kerberos i Win2k

- Implementasjonen er ganske lik RFC 1510
- Endringer:
  - Bruker ”Authorization Field”
  - Bruker TCP for store tickets
  - Har lansert autentisering av brukere via ”public-key certificates”

Authorization Field

- Win2k benytter Security Identifier (SID) for aksesskontrol; identifiserer en bruker
- En SID er unik i et enterprise, blir aldri brukt to ganger på forskjellige brukere
- En bruker kan være medlem i grupper, og en gruppe kan være medlem i andre grupper (nesting)
- En gruppe har også en SID

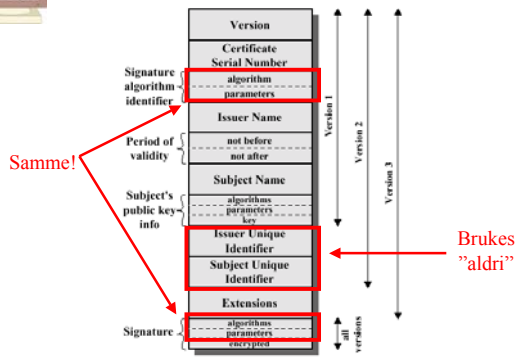
Authorization Field (forts.)

- Når en klient ber om en TGT, blir brukers SID + SID til alle grupper bruker tilhører lagt til i Authorization Field i TGT
- Når klienten ber om service ticket, blir Authorization Field i TGT kopiert over i ticket, og hvis server er i annet domain (realm), legges også inn eventuelle lokale grupper bruker er medlem av

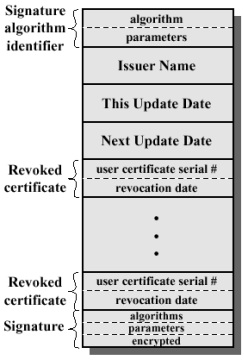
X.509

- Standard for sertifikater for offentlige nøkler
- Del av OSIs katalogtjeneste (X.500)
- Brukes i S/MIME, IPsec, SSL/TLS og SET

X.509 sertifikat



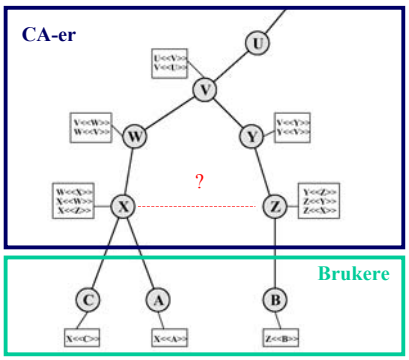
X.509 CRL



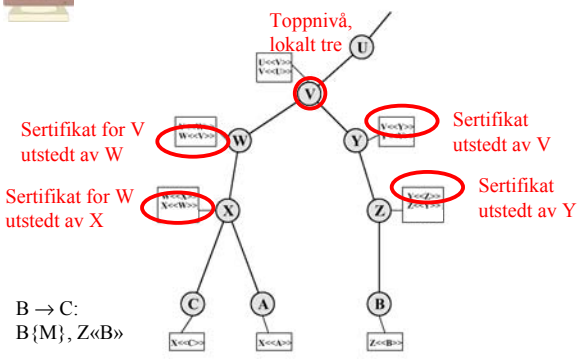
Notasjon

- Y<<X>>
  - Et sertifikat for bruker X utstedt av CA Y
  - CA<<A>>= CA{V,SN,AI,CA, T_A, A, Ap}
- Y{I}
  - Informasjon I *signert av* Y. Består av I og en "kryptert" hash av I (dvs.: I || $S_Y(H(I))$)
- CA_1<<CA_2>>
  - Sertifikat for CA_2 utstedt av CA_1.
  - Impliserer at CA_1 *går god for* CA_2

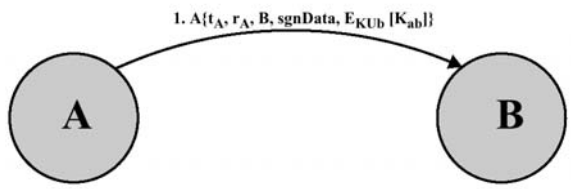
Hypotetisk CA-hierarki



CA eksempel

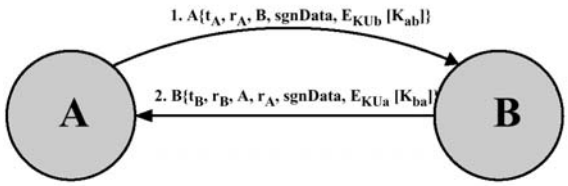


X.509 enveis autentisering



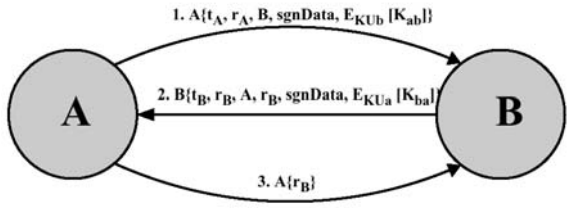
t_A – timestamp
r_A – nonce

X.509 toveis autentisering



t_B – timestamp fra B
r_B – nonce fra B

X.509 treveis autentisering



r_B – nonce fra B

Dagens website

- http://web.mit.edu/kerberos/www