Võtmete vahetamise algoritmid ja autentimine. Autentimine ja E-mail

Erika Matsak, PhD

Võtmete jagamise meetodid

- Krüptosüsteemi turvalisus sõltub sellest, kui turvaliselt on jagatud võtmed osapoolte vahel
  I. Võti on moodustatud A-ga ning “füüsiliselt” üle antud isikule B
  II. Võti on moodustatud kolmanda osapoole poolt ning on “füüsiliselt” ” üle antud A-le ja B-le
  III. A-l ja B-l on eelnevalt moodustatud ajutiselt kasutatav võti. Üks osapooltest saab vajadusel moodustada uut võtit vana võtme alusel ning edastada seda teisele osapoolele.
  IV. Kui A-l ja B-l on turvaline ühendus C-ga, siis C saab edastada võtit turvalise kanali kaudu nii A-le kui B-le.
- Esimest kaks meetodi nimetatakse “käsitsi” meetodiks võtmete vahetamiseks. Need on küll kõige turvalisemad, aga nende kasutamine on tihti päris ebamugav ning mõnikord isegi võimatu.

Võtmete vahetamise probleemid

- Esimese ja teise meetodi miinused:
  - Reaalsed süsteemid on hajutatud hostide ja serverite vahel.
  - Iga host peab omama võtme gruppi, mida toetatakse dünaamiliselt.
  - Võtmete arv sõltub sellest, mitu inimest peab osalema
  - Võrgus, mis koosneb N hostist on vaja [N (N - 1)]/2 võtit
  - Kui võtmeid vajatakse konkreetsete transaktsioonide jaoks, siis nende arv on tunduvalt suurem, sest iga transaktsioonipaar vajab oma võtit.
- Kolmanda meetodi miinused:
  - Kui ründaja saab kätte ühe võtme, siis ta saab tuletada ka kõik ülejäänud võtmed.
  - Võtmete arv on samuti suur
- Neljandat meetodit kasutatakse kaasaegsetes automatiseeritud süsteemides
  - Kasutatakse võtmete jagamise keskust (Key Destribution Centre - KDC), mis vastutab võtmete jagamise eest hostide ja protsesside vahel. Igal kasutajal peab olema oma unikaalne võti KDC –ga suhtlemiseks.

Võtmete jagamise keskus (Key Destribution Centre – KDC)

- Kasutatakse võtmete hierarhiat: master-võtmeid ja sessiooni võtmeid.
- Konfidentsiaalseks ühendamiseks kasutatakse ajutisi võtmeid, mis on oma loomu poolest sessiooni võtmed.
  - Iga sessiooni võti peab olema saadud võtmete jagamise keskuselt
  - Sessiooni võtmeid jagatakse krüpteeritud kujul, kasutades master-võtit
  - Sessiooni võtmete arv N kasutajale on [N (N - 1)]/2
- Master-võtme edastus peab olema samuti turvaline.
  - Selliseid võtmeid edastatakse kahe esimese meetodiga.
  - Master-võtmete arv on N

Sessiooni võtme kehtimise aeg

- Aeg, millal sessiooni võti kehtib võrdub sessiooni ajaga
- Mida tihedamini vahetatakse sessiooni võtmeid, seda vähem on aega ründajal võtme murdmiseks
- Teisest küljest: sessiooni võtmete genereerimine ning jaotus pidurdavad andmete vahetuse algust. Turvapoliitika peab tasakaalustama sessiooni aega ning valima optimaalseid väärtusi
- Kui sessiooni aeg on pikk, siis peab olema võimalik vahetada võtmeid sessiooni siseselt
- Kõige sagedamini kasutatakse sessiooni võtit, mis on fikseeritud ajaga ning fikseeritud transaktsioonide arvuga

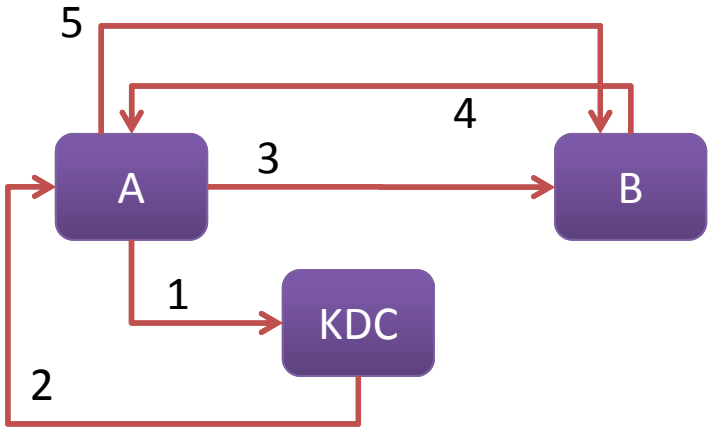
Autentimise protokollid

- Protokollide eesmärk on sessiooni võtmete jagamise konfidentsiaalsuse tagamine ning selle õigeaegsuse garanteerimine. Protokoll ei tohi lubada vana võtme korduvat kasutust.
- Kui õigeaegsus ei ole tagatud, siis on oht, et vastane saab kätte edastatava sessiooni võtme
- Replay-ründed:
  - Lihtne kordus: vastane kopeerib sessiooni võtme ning edastab selle hiljem veelkord
  - Vastane hävitab “praeguse” võtme ning saadab selle asemel eelnevalt kopeeritud võtme.
- Kui tegu oleks tavatekstiga, siis replay-rünnete vältimiseks saaks iga edastava teksti külge panna järjekorra numbri (sequence number). Uut teksti võetakse vastu ainult siis, kui selle järjekorra number on õige.
- Võtmete juures kasutatakse replay-rünnete vältimiseks järgmisi lahendusi:
  - Ajatempel. Osapool A võtab vastu võtme ainult siis, kui ajatemplis märgitud aeg vastab käesolevale hetkele. Osapoolte kellad peavad olema sünkroniseeritud (keeruline!).
  - Päring/vastus. Osapool A saadab päringus B-le juhuarvu (nonce - number only once) ning kontrollib, et vastus sisaldaks sama juhuarvu.

Needham-Schroederi protokoll

- Võtmete vahetamiseks ja autentimiseks kasutatakse tihti kahetasemelist hierarhiat, mis toetub sümmeetrilistele algoritmidele.
- Needham-Schroederi protokoll:
  1. $A \rightarrow KDC: ID_A \,||\, ID_B \,||\, N_1$
  2. $KDC \rightarrow A: E_{Ka}[K_S \,||\, ID_B \,||\, N_1 \,||\, E_{Kb}[K_S \,||\, ID_A]]$
  3. $A \rightarrow B: E_{Kb}[K_S \,||\, ID_A]$
  4. $B \rightarrow A: E_{KS}[N_2]$
  5. $A \rightarrow B: E_{KS}[f(N_2)]$

Roger Needham (1935-02-09 – 2003-03-01)



ID_A, ID_B- identifikaator A ja B jaoks
N1, N2- transaktsiooni identifikaator (*nonce*)
K_A – sümmeetriline võti KDC ja A vahel
K_S –sessiooni ühekordne võti
E_{Kb} – sümmeetriline võti KDC ja B vahel
f(N2) – funktsioon, mis modifitseerib N2

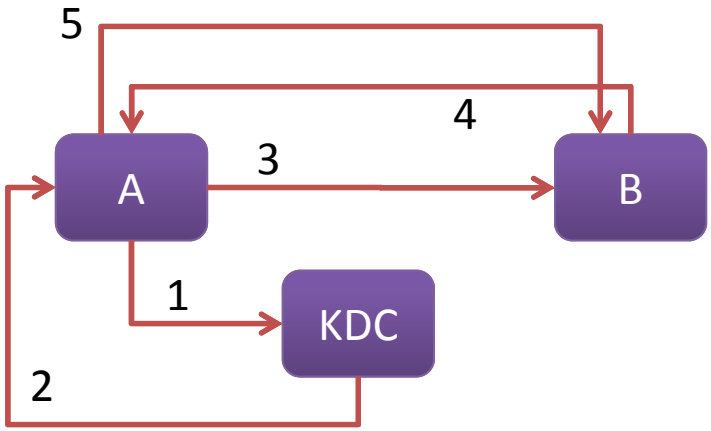
Needham-Schroederi protokoll

- Loetletud 5 sammu garanteerivad *B*-le, et sõnum, mis on saadud, ei ole muudetud ning ei ole eelmise sõnumi kordus.
- Sammud 1-3 vastutavad võtmete vahetamise eest. Sammud 3,4,5 autentimise eest.
- *A* saab võtme turvaliselt kätte sammul 2.
- Sõnum sammul 3 saab olla dešifreeritud ainult isiku *B* poolt.
- Samm 4 peegeldab seda, et *B* teab võtit K_S.
- Samm 5 garanteerib B-le, et A teab sessiooni võtit K_S ja kinnitab seda, et sõnum ei ole vana, kuna kasutatakse uut identifikaatorit N2.
- Sammud 4 ja 5 peavad kaitsma replay rünnete eest. Seejuures, kui vastane suudab üle võtta sõnumit sammul 3 ning korrata seda, siis see peaks kutsuma esile sessiooni katkemise. Risk seisneb selles, et kui vastane kordab sõnumit, mis on kätte saadud sammul 3 (see sisaldab sessiooni võtit) ning *B* ei mäleta kõikide eelmiste sessioonide võtmeid, siis vastane saab teeselda *A*-d.

Denning-Sacco protokoll

Täiendab Needham-Schroederi protokolli kasutades ajatempli sammudel 2 ja 3.

1. $A \rightarrow KDC: ID_A \,||\, ID_B$
2. $KDC \rightarrow A: E_{Ka} [K_S \,||\, ID_B \,||\, T \,||\, E_{Kb} [K_S \,||\, ID_A \,||\, T]\,]$
3. $A \rightarrow B: E_{Kb} [K_S \,||\, ID_A \,||\, T]$
4. $B \rightarrow A: E_{KS} [N_1]$
5. $A \rightarrow B: E_{KS} [f(N_1)]$



Denning-Sacco protokoll

Ajatempli kontrollimiseks saab kasutada valemi:
$|Clock - T| < \Delta t_1 + \Delta t_2$, kus
Δt_1 on KDC ja lokaalsete (*A* või *B*) kellade vahe ning
Δt_2 on võrgu võimalik viiteaeg.

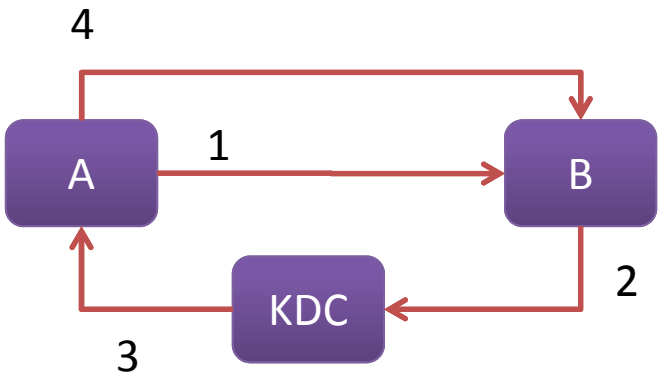
Ajatemplit šifreeritakse master-võtmega. Vastane, teades vana sessiooni võtit, ei suuda saavutada oma eesmärki, korrates sõnumit sammul 3 nõnda, et see ei oleks märgatav *B* jaoks ajalise nihke kujul.

Raskus on kellade sünkroniseerimises. Risk on seotud sellega, et kella saab rikkuda: oht ilmneb siis, kui saatja kell käib kiiremini võrreldes saaja kellaga. Sel juhul saab vastane selle vahe ära kasutada korduva sõnumi saatmiseks.

Pileti (ticket) kasutamine autentimiseks

- Antud protokoll annab võimaluse vältida probleeme, mis on seotud eespool nimetatud protokollidega.

1. $A \rightarrow B: ID_A \,||\, N_a$
2. $B \rightarrow KDC: ID_B \,||\, N_b \,||\, E_{Kb}\,[ID_A \,||\, N_a \,||\, T_b]$
3. $KDC \rightarrow A: EK_a\,[ID_B \,||\, N_a \,||\, K_S \,||\, T_b] \,||\, E_{Kb}\,[ID_A \,||\, K_S \,||\, T_b] \,||\, N_b$
4. $A \rightarrow B: E_{Kb}\,[ID_A \,||\, K_S \,||\, T_b] \,||\, E_{KS}\,[N_b]$



Identifikaator (nonce) N_a, mille A saadab B-le ning mille saab tagasi krüpteeritud kujul sammul 3, garanteerib A-le, et sessiooni võti ei ole vana.
Identifikaator N_b mängib sama rolli B jaoks.
T_b määrab millal sessiooni võti on vana
Pilet A jaoks on : $[ID_A \,||\, N_a \,||\, T_b]$

Pileti (ticket) kasutamine autentimiseks

- Protokoll autendib *A* ja *B* ning jagab sessiooni võtit.
- Protokoll annab *A*-le pileti, mille abil *A* saab autentida *B*-*d* ka korduvalt, ilma et peaks uuesti võtma ühendust serveriga KDC.
- Olgu, et *A* ja *B* vahel on antud protokolliga saadud ühendus. Kui vajalikud operatsioonid on sooritatud ning sessiooni aeg ei ole veel läbi, siis A saab luua uue sessiooni B-ga kasutades protokolli:

1. $A \rightarrow B: E_{Kb} [ID_A || K_S || T_b], N_a'$
2. $B \rightarrow A: N_b', E_{KS} [N_a']$
3. $A \rightarrow B: E_{KS} [N_b']$

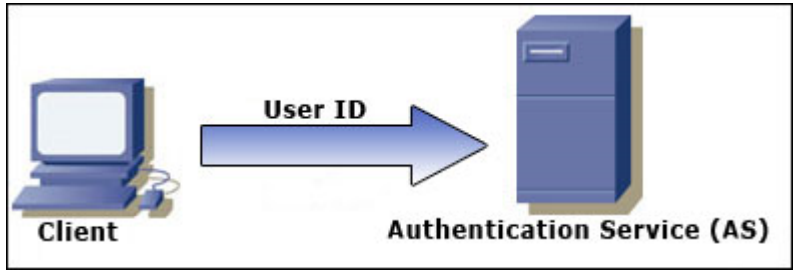
Sammul 1 kontrollitakse, et pilet ei oleks aegunud.
Uuesti moodustatud N_a' ja N_b' garanteerivat osapooltele, et ei olnud replay ründeid.
Aeg T_b on aeg lähtudes B kellast.
Seega ei vaja sünkroniseerimist.

Autentimine Kerberos protokolliga

- Väljatöötatud võrgu kaitseks klient-serveri tasemel projekti Athena käigus (Massachusetts Institute of Technology).
- Sõnumid, mis on edastatud Kerberose protokolliga, on kaitstud pealkuulamise ja replay rünnete eest.
- Oma loomu poolest on üks Needham-Schroederi protokollidest ning kasutab sümmeetrilist krüptosüsteemi. Algoritmi edasiarendus lubab ka avalike võtmete kasutamist autentimise juures.
- Kerberos 4 on väljatöötatud Steve Miller ja Clifford Neuman poolt 80ndate aastate lõpus
- Kerberos 5 on väljatöötatud John Kohl ja Clifford Neuman poolt 1993a ning seejärel edasiarendatud 2005a.
- 2007a on moodustatud Kerberos Consortium
  - Windows alates Windows 2000
  - Apple Mac OS X
  - Red Hat Enterprise Linux 4

Autentimine Kerberos protokolliga

- Protokoll opereerib krüpteeritud andmetega sümmeetriliste võtme abil.
- Protokoll koosneb kaheksast sammust



1. Autentimise servis saab kasutajalt päringu ning kontrollib, kas kasutaja on see kelleks end nimetab (lookup ID)

2. Peale kontrollimist lisatakse ajatempel, mis koosneb hetke kellajajast, mis näitab sessiooni algust ning kellaajast millal see aegub. Vaikimisi on sessiooni kehtivus 8 tundi. Samuti genereeritakse krüpteerimisvõti (kehtib samuti 8t)

3. Võti on saadetud kliendile tagasi koos piletiga.

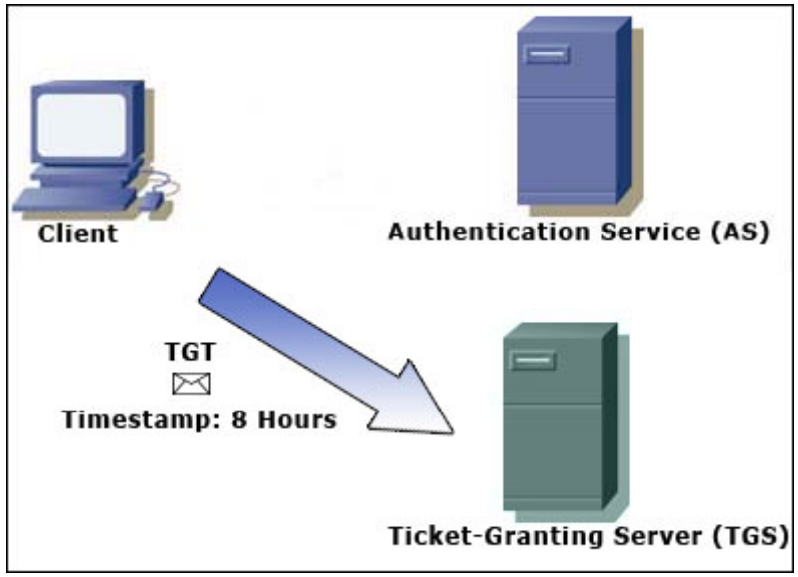
http://learn-networking.com/network-security/how-kerberos-authentication-works

Autentimine Kerberos protokolliga

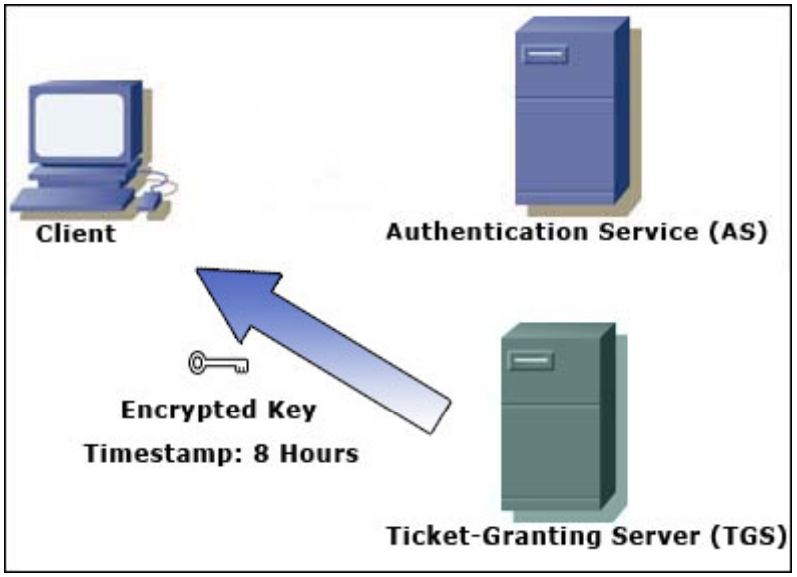
Vaatleme kolme esimest sammu lähemalt.

- Süsteemi sisenemine:
  - Kasutaja sisestab kasutajanime ja parooli kliendi masinas
  - Kliendi masin arvutab paroolist hash-funktsiooni ning tulemusest saab salastatud võti kliendi/kasutaja jaoks
- Kliendi autentimine:
  - Klient saadab serverile AS sõnumi, kus küsib luba kasutada teenuseid. NB! ei saadeta parooli ega võtit
  - AS kontrollib, kas on selline kasutaja baasis olemas. Kui on, siis saadetakse kliendile tagasi järgmisi asju:
    A. Sessiooni võti *klient/TGS*, mis on krüpteeritud kliendi salastatud võtmega
    B. TGT, mis sisaldab kliendi ID-d, kliendi võrgu aadressi, pileti kehtivuse aega, sessiooni võtit *klient/TGS*. TGT krüpteeritakse salastatud võtmega TGS.
  - Klient saab kaks sõnumit A ja B. Sõnumit A saab klient dešifreerida ning saab kätte sessiooni võtme. Sõnumit B klient dešifreerida ei saa.

Autentimine Kerberos protokolliga



4. Klient saadab pileti TGS serverisse, mis kontrollib seda piletit



5. TGS moodustab krüpteerimise võtme, varustab ajatempliga, kinnitab pileti ning saadab krüpteeritud kujul kliendile

http://learn-networking.com/network-security/how-kerberos-authentication-works

Autentimine Kerberos protokolliga

Sammud 4 ja 5 lähemalt.

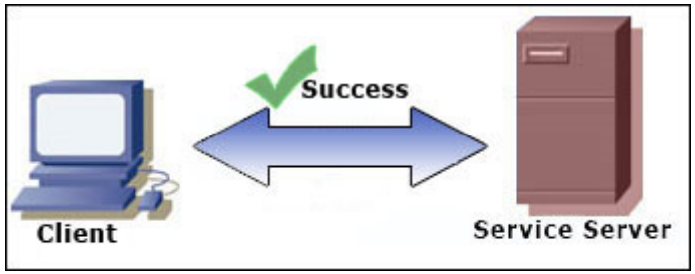
- Klient saadab TGS-le järgmised sõnumid:
  C. Sisaldab TGT-d, mis on saadud sõnumist B, ning nõutava teenuse ID-st
  D. Autentimise identifikaator, mis koosneb kliendi ID-st ja ajatemplist. See šifreeritakse sessiooni võtmega *klient/TGS.*
- Kui TGS saab kätte C ja D, siis TGS toob sõnumist C välja sõnumi B ning dešifreerib selle. Üks sõnumi komponentidest on sessiooni võti *klient/TGS.* Selle abil dešifreeritakse sõnumit D. Seejärel saadab TGS kliendile järgmisi sõnumeid:
  E. Pilet serveriga ühendamiseks (*Client-to-server ticket*), mis koosneb kliendi ID-st, kliendi võrgu aadressist, pileti kehtivusajast ning sessiooni võtmest *klient/server*. Pilet krüpteeritakse serveri SS salastatud võtmega
  F. Sessiooni võti *klient/server*, mis on krüpteeritud võtmega *klient/TGS*

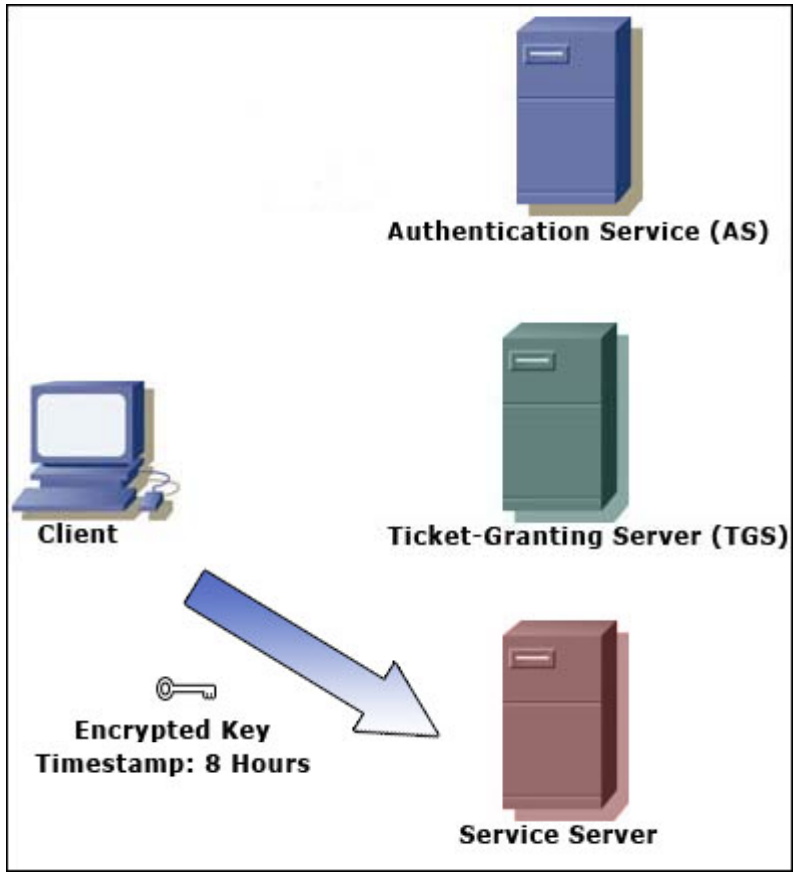
Autentimine Kerberos protokolliga

6. Klient dekrüpteerib pileti, annab sellest teada TGS-le. Seejärel klient saadab enda krüpteerimisvõtme serverisse (millega ongi vaja sessiooni luua).

7. Server dekrüpteerib võtme, kontrollib ajatempli kehtivust. Kui kehtib, siis Server loob sessiooni pileti

8. Klient dekrüpteerib pileti ning kui sessiooni võti kehtib, siis loob ühenduse serveriga





Autentimine Kerberos protokolliga

- Sammud 6,7 lähemalt:
  - Kui klient on saanud TGS-ilt sõnumid E ja F, siis tal on piisavalt informatsiooni, selleks et läbida autoriseerimine serveris SS. Klient võtab ühendust SS-ga ja saadab järgmised sõnumid:
    - Sõnum E, mis on pärit viiendast sammust, nimelt *klient/server* pilet, mis on krüpteeritud serveri SS salastatud võtmega

    G. Uus autentikaator (*authenticator*), mis koosneb kliendi ID-st ja ajatemplist, see krüpteeritakse *klient/server* sessiooni võtmega
  - SS dekrüpteerib pileti, kasutades enda SS salastatud võtit ning saab sealt kätte sessiooni võtme *klient/server.* Selle saadud võtmega dekrüpteerib G-d ning saab kätte autentikaatori. Seejärel saadab server kliendile järgmise sõnumi (mille abil seletab, et server on just server ja on valmis kliendiga suhtlema):

    H. Ajatempel, mis on esitatud kliendiga +1 ning šifreeritud sessiooni võtmega *klient/server* .

Autentimine Kerberos protokolliga

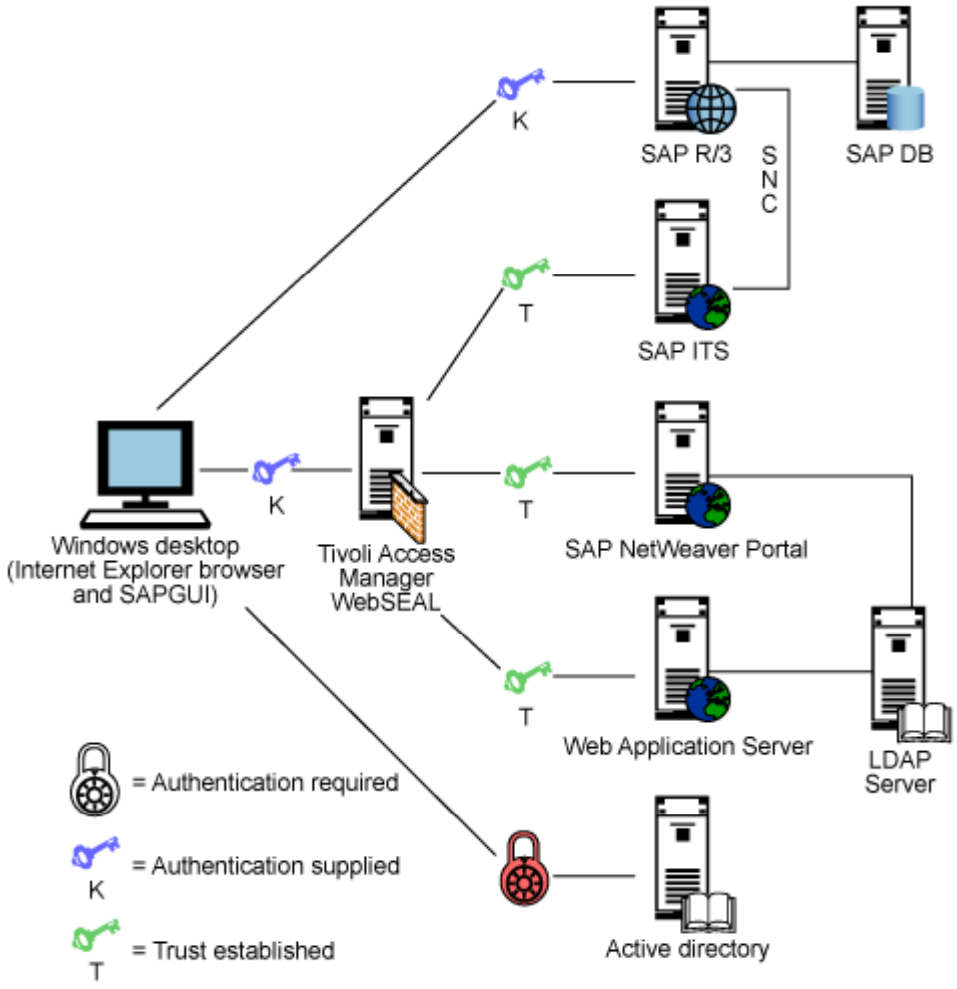
Samm 8 lähemalt:

- Klient dešifreerib H sessiooni võtmega *klient/server* ning kontrollib, kas ajatempel on korrektselt uuendatud. Kui on, siis klient saab serverit usaldada ning alustada päringute saatmist.
- Server osutab küsitud teenust

- Kõik sammud fikseeritakse logfailides

2005-05-29T22:22:27 TGS-REQ mike@HPPI.TROITSK.RU from IPv4:192.168.1.253 for ftp/kenga.hppi.troitsk.ru@HPPI.TROITSK.RU
2005-05-29T22:22:27 Server not found in database: ftp/kenga.hppi.troitsk.ru@HPPI.TROITSK.RU: No such entry in the database
2005-05-29T22:22:27 sending 145 bytes to IPv4:192.168.1.253
2005-05-29T22:22:27 TGS-REQ mike@HPPI.TROITSK.RU from IPv4:192.168.1.253 for host/kenga.hppi.troitsk.ru@HPPI.TROITSK.RU
2005-05-29T22:22:27 sending 550 bytes to IPv4:192.168.1.253
2005-05-29T22:22:28 TGS-REQ mike@HPPI.TROITSK.RU from IPv4:192.168.1.253 for krbtgt/HPPI.TROITSK.RU@HPPI.TROITSK.RU [forwarded, forwardable]
2005-05-29T22:22:28 sending 546 bytes to IPv4:192.168.1.253

Kerberos lahenduste näited



http://www.ibm.com/developerworks/tivoli/library/t-ssosaptam/index.html

Autentimine asümmeetriliste protokollidega

- Protokoll, mis kasutab ajatemplit ja autentimisserverit:

1. $A \rightarrow AS: ID_A \,||\, ID_B$
2. $AS \rightarrow A: E_{KRas}[ID_A \,||\, KU_a \,||\, T] \,||\, E_{KRas}[ID_B \,||\, KU_b \,||\, T]$
3. $A \rightarrow B: E_{KRas}[ID_A \,||\, KU_a \,||\, T] \,||\, E_{KRas}[ID_B \,||\, KU_b \,||\, T] \,||\, E_{KUb}[E_{KRa}[K_S \,||\, T]]$

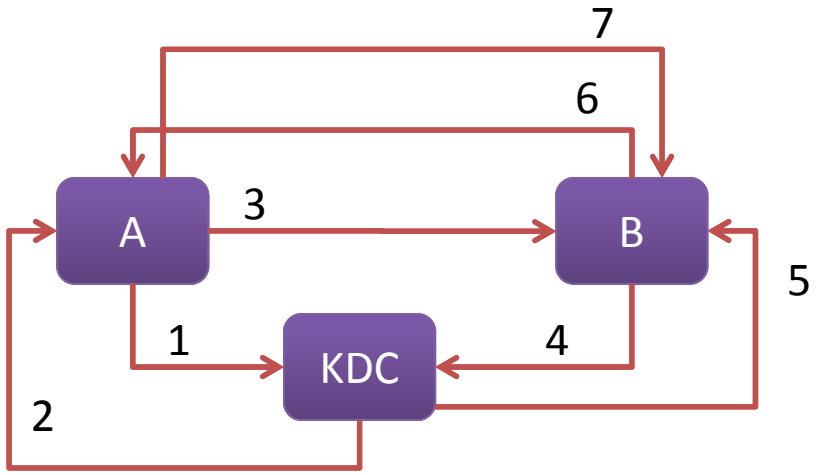
   Antud juhul on kolmanda osapoole rollis server AS. See server ei moodusta ega jaga võtmeid. Serveri ülesandeks on sertifitseerida osapoolte avalikke võtmeid.

   Osapool A valib ning krüpteerib sessiooni võtme K_s.

Autentimine asümmeetriliste protokollidega

- Autentimine mis kasutab KDC ja *nonce*.

1. $A \rightarrow KDC: ID_A \,||\, ID_B$
2. $KDC \rightarrow A: E_{KRauth} [ID_B \,||\, KU_b]$
3. $A \rightarrow B: E_{KUb} [N_a \,||\, ID_A]$
4. $B \rightarrow KDC: ID_B \,||\, ID_A \,||\, E_{KUauth} [N_a]$
5. $KDC \rightarrow B: E_{KRauth} [ID_A \,||\, KU_a] \,||\, E_{KUb} [E_{KRauth} [N_a \,||\, K_S \,||\, ID_A \,||\, ID_B]]$
6. $B \rightarrow A: E_{KUa} [E_{KRauth} [N_a \,||\, K_S \,||\, ID_A \,||\, ID_B] \,||\, N_b]$
7. $A \rightarrow B: E_{KS} [N_b]$



Ühesuunaline autentimine ja e-mail

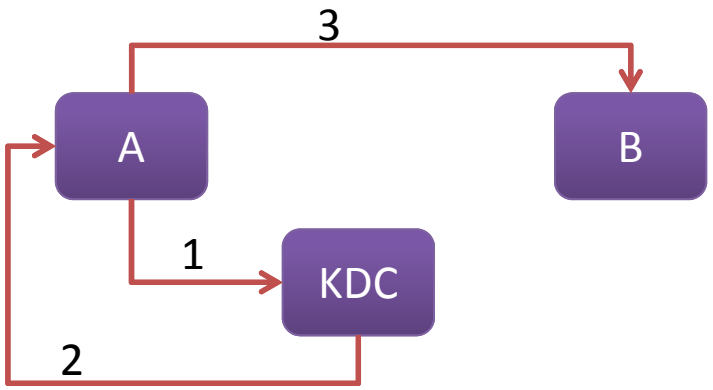
- E-maili puhul ei pea saatja ning saaja olema üheaegselt ühenduses (ei looda sessiooni). Kirju hoitakse postkastis.
- Saatja ning saaja tahavad olla kindlad, et kiri on jõudnud kohale tervikuna ning ei ole võltsitud.
- Saatja tahab olla kindel, et kiri on jõudnud adressaadile.
- Saaja tahab garantiid, et kiri on tulnud just sellelt inimeselt, kes on märgitud saatjaks. Nimetatud garantii puudumise üheks tagajärjeks on SPAM.

- SPAM-iga võitlemisks:
  - SPF (Sender Policy Framework) [http://www.openspf.org/]
  - SenderID [http://www.microsoft.com/mscorp/safety/technologies/senderid/default.mspx]
  - DomainKeys Identified Mail. Autentimine, mis kasutab digiallkirja [http://www.dkim.org/]

Ühesuunaline autentimine ja e-mail. Kolmanda osapoole KDC kasutamine

Lahendus sümmeetrilise krüptoalgoritmiga

1. $A \rightarrow KDC: ID_A \,||\, ID_B \,||\, N_1$
2. $KDC \rightarrow A: E_{Ka}[K_S \,||\, ID_B \,||\, N_1 \,||\, E_{Kb}[K_S \,||\, ID_A]]$
3. $A \rightarrow B: E_{Kb}[K_S, ID_A] \,||\, E_{KS}[M]$



Antud meetod garanteerib, et ainult õige saaja saab seda kirja lugeda. Teatud kindlusega garanteerib, et saatjaks on just isik A.
Protokoll ei kaitse replay-rünnete eest.
Saab kasutada ajatemplit, kuigi mitteefektiivselt (loomulikud viivitused e-maili edastamisel).

Ühesuunaline autentimine ja e-mail. Kolmanda osapoole KDC kasutamine

Lahendus asümmeetrilise krüptoalgoritmiga

Konfidentsiaalsuse tagamine:

$A \rightarrow B: E_{KUb} [K_S] || E_{KS} [M]$

Teksti šifreeritakse ühekordse salastatud võtmega K_s. Ning kogu pakend šifreeritakse *B* isiku avaliku võtmega.

Autentimine, mis sisaldab digiallkirjastamist:

$A \rightarrow B: M || E_{KRa} [H (M)]$

Antud meetod garanteerib, et A ei saa tagantjärgi väita, et kiri pole temalt saadetud.

Riskid: Vastane saab enne kirja edastamist (kiri seisab järjekorras) välja lõigata õige allkirja, panna selle asemele võltsitud allkirja ning panna kirja tagasi järjekorda. Lahendus:

krüpteerida sümmeetrilise võtmega Ks, see edastatakse krüpteerides avaliku võtmega

$A \rightarrow B: E_{KS} [M || E_{KRa} [H (M)]] || E_{KUb} [K_S]$

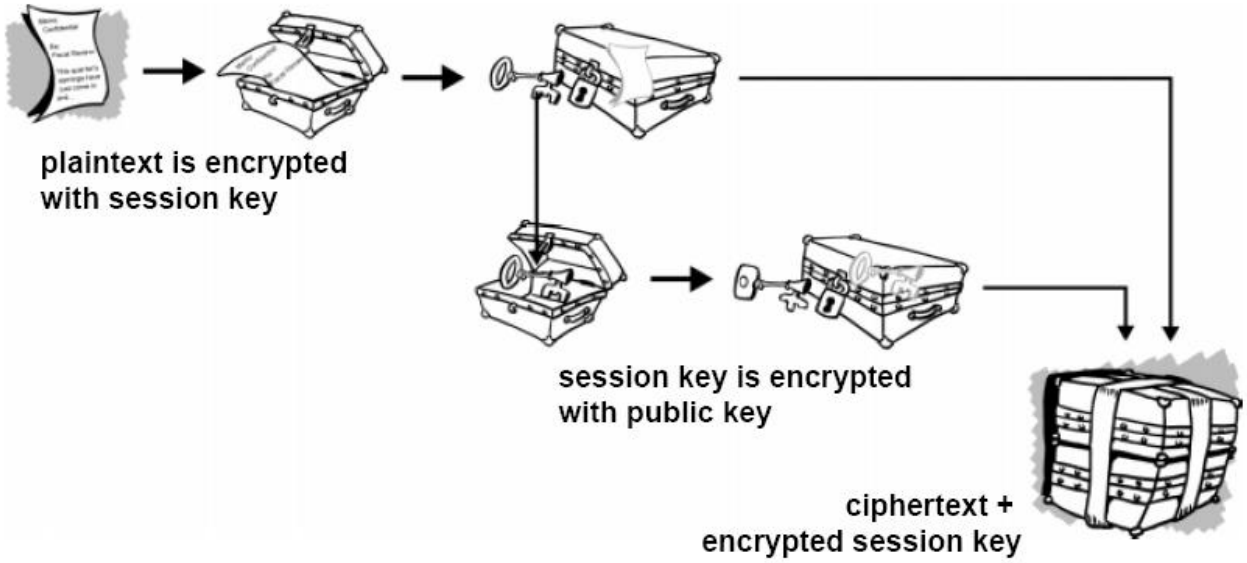
- Teine variant: kasutada autentimisserveri privaatvõtit KRas ning sertifikaati

  $A \rightarrow B: M || E_{KRa} [H (M)] || E_{KRas} [T || ID_A || KU_a]$

E-maili krüpteerimise levinumad protokollid. PGP

- PGP (Pretty Good Privacy) töötas välja Philip Zimmermann, 1991a http://philzimmermann.com/EN/background/index.html

Phil Zimmermann, 12.02.1954



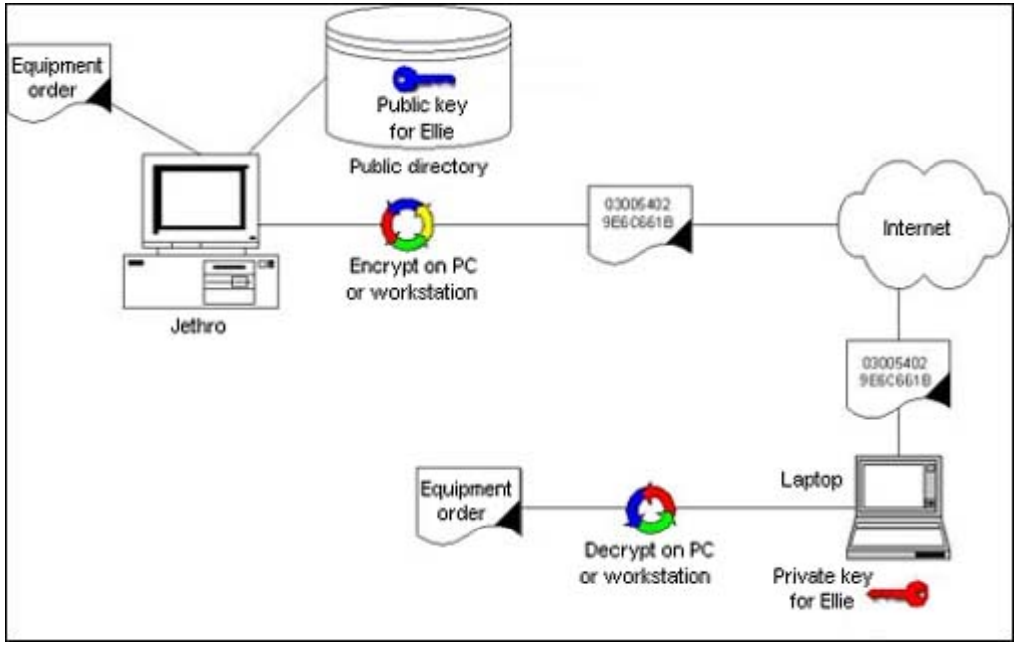
How PGP encryption works

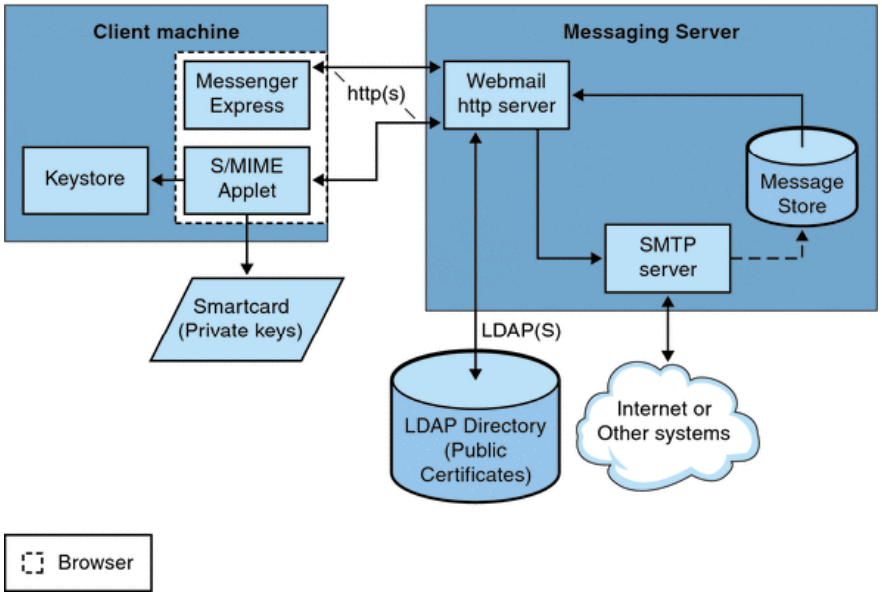
Võtme genereerimise juures antakse ette omaniku nimi ja e-mail, võtme tüüp, pikkus ja kehtivusaeg.
Võtme tüübid:
RSA v4, RSA legacy (v3) ja Diffie-Hellman/DSS

http://www.data-processing.hk/glossaries/pgp/

E-maili krüpteerimise levinumad protokollid. S/MIME

- S/MIME sisaldab autentimist, terviklikuse kontrollimist, krüpteerimist.





http://www.ibm.com/developerworks/lotus/library/securemessaging/
http://docs.sun.com/app/docs/doc/819-4428/bgbcq?a=view