Allekirjoitetut sertifikaatit

- **Julkisen avaimen protokollissa keskeistä on julkisten avainten aitous.**
- Alice voi välittää Bobille luotetun osapuolen allekirjoittaman sertifikaatin, joka todistaa hänen julkisen avaimensa aitouden.

1.	$A \longrightarrow T:$	A, B
2.	$T \longrightarrow A:$	$T_T, B, K_B^+,\ S_T(T_T, B, K_B^+)$
		$T_T, A, K_A^+,\ S_T(T_T, A, K_A^+)$
3.	$A \longrightarrow B:$	$T_T, A, K_A^+,\ S_T(T_T, A, K_A^+)$
		$E_B(T_A, B, K_{SES},\ S_A(T_A, B, K_{SES}))$

Sertifikaatit voivat olla pitkäikäisiä, jolloin niitä ei tarvitse aina välittää uudestaan.

$$S_T(T_T, L, A, K_A)$$

A = käyttäjä

K_A = julkinen avain

T_T, L = luontiaika ja voimassaoloaika

Uusi istuntoavain on helppo luoda vanhojen sertifikaattien ollessa voimassa.

X.509-sertifikaatti

- CCITT/ISO 1988, ITU 1993
- Standardi määrittelee
  - autentikoinnissa käytettävän sertifikaatin muodon
  - useiden sertifiointiauktoriteettien yhteistoiminnan
  - kolme autentikointiprotokollaa
  - sertifikaattien peruuttamisen.
- Standardi suosittelee RSA:ta, mutta sallii eri salaus- ja allekirjoitusalgoritmit.
- Sertifikaatin muoto

$$CA << A >> = S_{CA}(SN, CA, T_{CA}, L, A, K_A)$$

- Sertifikaatin antaa **sertifiointiauktoriteetti CA** (*certification authority*) .
- Sertifikaatilla käyttäjä todistaa toiselle samaan CA:han luottavalle käyttäjälle julkisen avaimensa aitouden.
- Sertifikaatit voivat olla julkisessa palvelimessa kaikkien saatavilla.

X.509-CA-hierarkia

- CA:t muodostavat hierarkian, jossa jokainen sertifioija jakaa puussa ylä- ja alapuolella olevien naapureiden sertifikaatteja.

  Alice voi todistaa Bobille avaimensa aitouden, jos Alicen ja Bobin välillä on **sertifikaattipolku**.

$$CA_A << CA_1 >> CA_1 << CA_2 >> \\ CA_2 << CA_3 >> \ldots CA_N << B >>$$

- Alice luottaa CA_A:han ja transitiivisesti tämän yläpuolella oleviin auktoriteetteihin.

  Jokin auktoriteetti CA' on puussa sekä Alice attä Bobin yläpuolella.

  Alice luottaa CA':un ja kaikkiin tämän alapuolella puussa oleviin auktoriteetteihin.
- **Huom! Yleisessä tapauksessa luottamus ei ole transitiivista.** Kuinka syviä hierarkioita sallitaan?

X.509-sertifikaatin peruutus

(*revocation of certificates*)

- Sertifikaatti peruutetaan, kun aikaleima on vielä käypä ja
  - yksityinen avain on paljastunut
  - käyttäjään tai sertifiointiauktoriteettiin ei enää luoteta
  - CA ei enää halua sertifioida käyttäjää.
- Lista peruutetuista sertifikaateista on kaikilla CA:lla ja käyttäjillä muistissa.

  Listassa on sertifikaatin numero.
- Käyttäjä päivittää listaansa haluamassaan tahdissa.
- Peruutuslistojen ylläpito on kallista. Vaihtoehtona **lyhyet voimassaoloajat ja sertifikaattien säännöllinen päivitys**.

X.509-autentikointi

- Autentikoinnin vaiheet:
  1. Osapuolet hankkivat sertifikaattipolun toisiinsa.
  2. Autentikointi ja avaintenvaihto tapahtuu julkisia avaimia käyttäen.
- Yleensä Alicella ja Bobilla on muistissa sertifikaattipolku puussa itsestä ylöspäin.

  Kumpikin lähettää oman polkunsa toiselle tai sertifikaatit haetaan palvelimesta.
- **Yksisuuntainen X.509-autentikointi**

$$\begin{array}{lll} 1. & A \longrightarrow B: & T_A, N_A, B, E_B(K_{AB}), \\ & & S_A(T_A, N_A, B, E_B(K_{AB})) \end{array}$$

  - A lähetti sanoman ja se on tarkoittu B:lle.
  - Sanomaa ei ole muutettu tai toistettu.
  - Vaatii globaalin kellon. Aikaleima T_A sisältää voimassaolon alkuajan ja loppuajan.
  - B voi muistaa nonssin kunnes aikaleima vanhenee, jotta sanomaa ei voi toistaa sinä aikana.

- **Heikkous 1**:

  Salatun X.509-istuntoavaimen voi kopioida sitä tuntematta.

$$\begin{array}{ll} A \longrightarrow B: & T_A, N_A, B, E_B(K_{AB}), \\ & S_A(T_A, N_A, B, E_B(K_{AB})) \\ M \longrightarrow B: & T_M, N_M, B, E_B(K_{AB}), \\ & S_M(T_M, N_M, B, E_B(K_{AB})) \end{array}$$

  Kun istunnoissa on sama avain, voi hyökkääjä kopioida sanomia istuntojen välillä.

  Esim: A lähettää B:lle ja M:lle saman tenttikysymyksen. M kopioi B:n vastauksen näkemättä sitä.

$$\begin{array}{ll} A \longrightarrow B: & \text{"Mikä on nonssi?"} \\ B \longrightarrow A: & E_{AB}(\text{"Nonssi on satunnaisluku."}) \\ M \longrightarrow A: & E_{AB}(\text{"Nonssi on satunnaisluku."}) \end{array}$$

- **Kaksisuuntainen X.509-autentikointi**

$$\begin{array}{lll} 1. & A \longrightarrow B: & T_A, N_A, B, E_B(K_{AB}) \\ & & S_A(T_A, N_A, B, E_B(K_{AB})) \\ 2. & B \longrightarrow A: & T_B, N_B, A, N_A, E_A(K_{BA}) \\ & & S_B(T_B, N_B, A, N_A, E_A(K_{BA})) \end{array}$$

  - B vastasi ja vastaus on tarkoittu A:lle.
  - Sanomaa ei ole muutettu tai toistettu.
  - T_B on tarpeeton. Sanoman 1 tuoreus riippuu edelleen globaalista kellosta.
- Istuntoavaimia on kaksi. **Liikenne eri suuntiin salataan eri avaimella.**

  Hyöty: dataa voi salata etukäteen puskuriin tai lähettää samalla avaimella salattuna useammalle. Toistohyökkäys vaikeutuu.

  Huom: Avaimen voi yhä kopioida.

  (Avainmateriaalit voidaan myös yhdistää yhdeksi avaimeksi.)

- **Kolmisuuntainen X.509-autentikointi**

  CCITT/ITU-standardi (1988-1993)

$$\begin{array}{lll} 1. & A \longrightarrow B: & T_A, N_A, B, E_B(K_{AB}) \\ & & S_A(T_A, N_A, B, E_B(K_{AB})) \\ 2. & B \longrightarrow A: & T_B, N_B, A, N_A, E_A(K_{BA}) \\ & & S_B(T_B, N_B, A, N_A, E_A(K_{BA})) \\ 3. & A \longrightarrow B: & N_B,\ S_A(N_B) \end{array}$$

  - A on saanut avaimen K_{BA}.
  - Aikaleimat ja globaali kello tarpeettomia. Aikaleimat voivat saada arvon $T_A = T_B = 0$, eikä niitä tarvitse tarkistaa.
  - Sanoman 3 pitäisi osoittaa B:lle sanoman 1 tuoreus. Sanoma 2 sitoo nonssit R_A ja R_B yhteen.

- **Heikkous 2**:

  Vanhan istuntoavaimen haltuunsa saanut hyökkääjä voi siirtää X.509-autentikoinnin eteenpäin (BAN89). (Aikaleimoja ei käytetä.)

$$
\begin{array}{lll}
A \longrightarrow B: & 0, N_A, B, E_B(K_{AB}), & \\
 & S_A(0, N_A, B, E_B(K_{AB})) & \\
\ldots & & \\
A \longrightarrow M: & 0, N'_A, M, E_M(K_{AM}), & (1') \\
 & S_A(0, N'_A, M, E_M(K_{AM})) & \\
M(A) \longrightarrow B: & 0, N_A, B, E_B(K_{AB}), & (1) \\
 & S_A(0, N_A, B, E_B(K_{AB})) & \\
B \longrightarrow M(A): & 0, N_B, A, N_A, E_A(K_{BA}), & (2) \\
 & S_B(0, N_B, A, N_A, E_A(K_{BA})) & \\
M \longrightarrow A: & 0, N_B, A, N'_A, E_A(K_{MA}), & (2') \\
 & S_M(0, N_B, A, N'_A, E_A(K_{MA})) & \\
A \longrightarrow M: & N_B,\ S_A(N_B) & (3') \\
M(A) \longrightarrow B: & N_B,\ S_A(N_B) & (3)
\end{array}
$$

- Korjattu X.509 (ITU 1993):

$$
\boxed{
\begin{array}{lll}
1. & A \longrightarrow B: & T_A, N_A, B, E_B(K_{AB}) \\
 & & S_A(T_A, N_A, B, E_B(K_{AB})) \\
2. & B \longrightarrow A: & T_B, N_B, A, N_A, E_A(K_{BA}) \\
 & & S_B(T_B, N_B, A, N_A, E_A(K_{BA})) \\
3. & A \longrightarrow B: & N_B, B\ S_A(N_B, B)
\end{array}
}
$$

- Huom: Avaimen voi yhä kopioida istunnosta toiseen.

  Mahdollinen korjaus: lasketaan varsinaiset istuntoavaimet yksisuuntaisella funktiolla avainmateriaaleista K_{AB} ja K_{BA} sekä osapuolten nimistä.

$$K'_{AB} = HASH(A, B, K_A^+, K_B^+, K_{AB}).$$

Kerberos

- MIT, DEC
- Kerberos V5 1993, RFC1510
- Ei pelkästään avaintenjakeluun, vaan palvelinresurssien hallintaan.
- Eri palvelimissa:
  1. autentikointi (*authentication server AS*)
  2. oikeuksien hallinta (*ticket granting server TGS*)
  3. varsinainen palvelu (*server V*) .
- Käyttää symmetristä salausta. Jokainen käyttäjä jakaa autentikointipalvelimen kanssa salaisen avaimen.

Kerberos: perusrakenne

- Kerberos 5-protokollan perusrakenne:

  T = tikettipalvelin

$$
\boxed{
\begin{array}{lll}
1. & A \longrightarrow T: & A, B \\
2. & T \longrightarrow A: & E_A(T_T, B, K_{AB}), E_B(T_T, A, K_{AB}) \\
3. & A \longrightarrow B: & E_B(T_T, A, K_{AB})
\end{array}
}
$$

  Sertifikaatille pidempi käyttöikä ja Alice todistaa läsnäolonsa Trentille:

$$
\boxed{
\begin{array}{lll}
1. & A \longrightarrow T: & A, B \\
2. & T \longrightarrow A: & E_A(T_T, B, K_{AB}), E_B(L, A, K_{AB}) \\
3. & A \longrightarrow B: & E_B(L, A, K_{AB}), E_{AB}(T_A)
\end{array}
}
$$

  L = voimassaoloaika (alku- ja loppuaika)

- RFC käyttää nonsseja, implemetaatiossa nonsseina aikaleimoja

C = *client*

AS = *authentication server*

TGS = *ticket granting server*

V = *server*

1.	$C \longrightarrow AS:$	$C,\ TGS$
2.	$AS \longrightarrow C:$	$E_C(T_{AS}, TGS, K_{C,TGS}),$ $TicketGrantingTicket$
3.	$C \longrightarrow TGS:$	$V,\ TicketGrantingTicket,$ $Authenticator_{C1}$
4.	$TGS \longrightarrow C:$	$E_{C,TGS}(T_{TGS}, V, K_{C,V}),$ $Ticket$
5.	$C \longrightarrow V:$	$Ticket,\ Authenticator_{C2}$
6.	$V \longrightarrow C:$	$E_{C,V}(T_{C2})$

1.	$C \longrightarrow AS:$	C, TGS
2.	$AS \longrightarrow C:$	$E_C(T_{AS},\ TGS,\ K_{C,TGS}),$ $E_{TGS}(L_1,\ C,\ K_{C,TGS})$
3.	$C \longrightarrow TGS:$	$V,\ E_{TGS}(L_1, C, K_{C,TGS}),$ $E_{C,TGS}(T_{C1})$
4.	$TGS \longrightarrow C:$	$E_{C,TGS}(T_{TGS},\ V,\ K_{C,V}),$ $E_V(L_2,\ C,\ K_{C,V})$
5.	$C \longrightarrow V:$	$E_V(L_2,\ C,\ K_{C,V}),\ E_{C,V}(T,\ data)$
6.	$V \longrightarrow C:$	$E_{C,V}(T_{C2})$

Diffie-Hellman

- Alice haluaa yhteisen istuntoavaimen Bobin kanssa. Alice ja Bob tietävät toistensa julkiset avaimet.
- Diffie–Hellman takaa avaimen tuoreuden, eikä vaadi kelloa:

1.	$A \longrightarrow B:$	$A, S_A(\text{"DH 1"}, DH_A)$
2.	$B \longrightarrow A:$	$B, S_B(\text{"DH 2"}, DH_B)$

Alice voi myös lähettää samalla sertifikaatin:

1.	$A \longrightarrow B:$	$A, S_A(\text{"DH 1"}, DH_A), S_T(T_T, L, A, K_A)$
2.	$B \longrightarrow A:$	$B, S_B(\text{"DH 2"}, DH_B)$