Avaintenhallinta

- Avain on kryptografisten järjestelmien keskeinen osa.
- Avainten pitää olla **aitoja** ja salaisten avainten **salaisia**.
- Avaimia välitetään **avaintenhallintaprotokollilla**.
- Kryptografisten protokollien suunnittelu on yhtä vaikeaa kuin algoritmienkin. Kannattaa käyttää valmiita, hyväksi tunnettuja protokollia.
- Avaintenvaihto, avaintenjakelu (*key establishment/agreement/exachange/ distribution/transport*)

  Protokollat käyttävät sekä salaisen että julkisen avaimen algoritmeja.

Istuntoavain

- Istuntoavain on (yleensä) salainen avain, jota käytetään vain **yhden istunnon ajan istunnon sisällön suojaamiseen.**

  Istuntoavaimista sovitaan tai ne jaetaan erityisillä avaintenjakeluprotokollilla.

  Istuntoavaimia voi olla useita eri tarkoituksiin, esim. salaukseen ja integriteettitarkistukseen eri suuntaisilla kanavilla.

  Istuntoavain generoidaan ja/tai välitetään salaisen **pääavaimen** tai julkisten avainten avulla.
- Istuntoavaimia tarvitaan, koska
  - tarvitaan jaettu salainen avain, eikä sitä ennestään ole
  - tarvitaan toistuvasti uusia avaimia tai useita avaimia eri tarkoituksiin
  - salataan suuria tietomääriä, jolloin kryptoanalyysi helpottuu.
- Istuntoavaimen paljastuminen ei saa vaarantaa pääavainta.

  Vanhojen istuntoavaimien paljastuminen ei saa vaarantaa uusia eikä uusien vanhoja istuntoavaimia.

  Parhaassa tapauksessa pääavaimelle ei ole saatavilla tunnettua selväkieltä.

Yksinkertainen yritelmä

- Alice haluaa yhteisen istuntoavaimen Bobin kanssa.

1. $A \rightarrow B:$ A, K_{SES}

  Hyvä?
- Salataan istuntoavain Alicen ja Bobin välisellä **pääavaimella** (*master key*) .

1. $A \rightarrow B:$ $A, E_{AB}(K_{SES})$

Avaintenhallinnan ensimmäinen tavoite on salaisten avainten salassa pysyminen. Se saavutetaan salauksella.

Joko kelpaa?

Sanomien rakennusosat

A, ID_A, B, T	protokollaosapuolien nimet tai tunnisteet
R_A	A:n generoima satunnaisluku
N_A	A:n generoima nonssi
T_A	A:n aikaleima
K_{AB}, K_{SES}	istuntoavain
K_{AT}	A:n ja T:n välinen pääavain
K_A	A:n julkinen, yksityinen tai salainen avain
K_A^+, K_A^-	A:n julkinen ja yksityinen avain
$E_K(M)$	salaus avaimella K
$E_{AB}(M)$	salaus avaimella K_{AB}
$E_A(M)$	salaus A:n julkisella tai salaisella avaimella
$MAC_K(M)$	avaimella K laskettu MAC
$S_A(M)$	A:n allekirjoitus
$h(M)$	yksisuuntainen funktio

- Allekirjoitus liitetään sanoman perään $M, S_A(M)$ tai lähetetään erikseen $S_A(M)$.

  Allekirjoitettua sanomaa merkitään myös $S_A(M)$ tai $\{M\}_A$.
- Salausta merkitään monin tavoin:

  $E_{K_{A,B}}(sanoma) = E_{AB}(sanoma) = E_{K_{AB}}[sanoma] = E(K_{A,B}; sanoma) = (sanoma)_{K_{A,B}} = K_{A,B}(sanoma).$

  Salauksen oletetaan usein sisältävän MAC:in.

Wide-mouth-frog

(Oletetaan salauksen E estävän sanoman hallitun muuntelun.)

1.	$A \to T:$	$A, E_A(T_A, B, K_{SES})$
2.	$T \to B:$	$E_B(T_T, A, K_{SES})$

Schneier ei näe protokollassa suurta vikaa. Entä sinä?

- Toistohyökkäys *Wide-Mouth Frog*-protokollaa vastaan:

$$
\begin{aligned}
A \to T: &\quad A, E_A(T_A, B, K_{SES}) \\
T \to B: &\quad E_B(T_T, A, K_{SES}) \\
M(B) \to T: &\quad B, E_B(T_T, A, K_{SES}) \\
T \to M(A): &\quad E_A(T_{T2}, B, K_{SES}) \\
M(A) \to T: &\quad A, E_A(T_{T2}, B, K_{SES}) \\
T \to M(B): &\quad E_B(T_{T3}, A, K_{SES}) \\
M(B) \to T: &\quad B, E_B(T_{T3}, A, K_{SES}) \\
T \to M(A): &\quad E_B(T_{T4}, B, K_{SES}) \\
&\dots \\
M \to T: &\quad A, E_B(T_{T1234}, A, K_{SES}) \\
T \to B: &\quad E_B(T_{T1234}, A, K_{SES})
\end{aligned}
$$

Mallory voi pitää avaimen K_{SES} elossa mielivaltaisen pitkään.

Salaisen avaimen Needham-Schröder

- Needham–Schröder-protokolla (1978–1981) oli ensimmäinen julkaistu salausta käyttävä avaintenhallintaprotokolla.

  (Oletetaan salauksen E estävän sanoman hallitun muuntelun.)

1.	$A \to T:$	A, B, N_A	
2.	$T \to A:$	$E_A(N_A, B, K_{SES}, E_B(K_{SES}, A))$	
3.	$A \to B:$	$E_B(K_{SES}, A)$	
4.	$B \to A:$	$E_{K_{SES}}(N_B)$	haaste
5.	$A \to B:$	$E_{K_{SES}}(N_B + 1)$	vaste

Sanomilla 4 ja 5 Bob varmistuu, että Alice tuntee avaimen (*key confirmation*) .

- Toistohyökkäys: Mallory nauhoittaa sanoman $E_B(K_{SES}, A)$. Myöhemmin hän saa haltuunsa vanhan istuntoavaimen K_{SES}.

$$
\begin{aligned}
M(A) \to B: &\quad E_B(K_{SES}, A) \\
B \to M(A): &\quad E_{K_{SES}}(N'_B) \\
M(A) \to B: &\quad E_{K_{SES}}(N'_B + 1)
\end{aligned}
$$

Denning–Sacco

- Denning–Sacco-protokollassa (1981–1994) Alice pyytää Trentiltä sertifikaatin omalle ja Alicen julkiselle avaimelle.

  Käytetään julkisen avaimen algoritmeja.

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

Normaalin protokolla-ajon jälkeen Bob jatkaa:

$$
\begin{aligned}
B \to T: &\quad B, C \\
T \to B: &\quad C, K_C, S_T(C, K_C),\ B, K_B, S_T(B, K_B) \\
B(A) \to C: &\quad E_C(T_A, K_{SES}, S_A(T_A, K_{SES})), \\
&\qquad C, K_C, S_T(C, K_C),\ A, K_A, S_T(A, K_A)
\end{aligned}
$$

Charlie luulee nyt, että K_{SES} on hänen ja Alicen yhteinen avain. Tosiasiassa se on Bobin ja Charlien välinen.

Myöskään sertifikaattien tuoreus ei ole selvää.

Meyer–Matyas

- Meyer–Matyas-protokollassa palvelin kertoo pyynnöstä julkisen avaimen. Avain liitetään pyyntöön nonssilla. Yksinkertaistettu versio:

1.	$A \to T:$	A, B, N_A
2.	$T \to A:$	$K_B, N_A,\ S_T(K_B, N_A)$

- Henkilön nimeä ei ole kunnolla liitetty avaimeen.

$$
\begin{aligned}
A \to T: &\quad A, B, N_A \\
T \to M(A): &\quad K_B, N_A,\ S_T(K_B, N_A) \\
M(A) \to T: &\quad A, M, N_A \\
T \to A: &\quad K_M, N_A,\ S_T(K_M, N_A)
\end{aligned}
$$

Mallory voi esiintyä Alicelle Bobina. Jos hän myös esiintyy Bobille Alicena, on kyseessä *man-in-the-middle*-hyökkäys.

Julkisen avaimen Needham-Schröder

- Julkisen avaimen NS-protokollassa (1978–1995) Alice ja Bob hankkivat toistensa julkiset avaimet Trentiltä ja vaihtavat avainmateriaalia N_A, N_B.

1.	$A \longrightarrow T$	A, B
2.	$T \longrightarrow A$	$K_B^+, B,\ S_T(K_B^+, B)$
3.	$A \longrightarrow B$	$E_B(N_A, A)$
4.	$B \longrightarrow T$	B, A
5.	$T \longrightarrow B$	$K_A^+, A,\ S_T(K_A^+, A)$
6.	$B \longrightarrow A$	$E_A(N_A, N_B)$
7.	$A \longrightarrow B$	$E_B(N_B)$

Istuntoavain voidaan generoida nonsseista: $K_{SES} = h(N_A, N_B)$.

Heikkous 1: Trentiltä saadut sertifikaatit voivat olla vanhoja.

- Oletetaan Alicen ja Bobin jo tuntevan toistensa julkiset avaimet:

1.	$A \longrightarrow B$	$E_B(N_A, A)$
2.	$B \longrightarrow A$	$E_A(N_A, N_B)$
3.	$A \longrightarrow B$	$E_B(N_B)$

Heikkous 2: Hyökkääjä pystyy siirtämään A:n autentikoinnin B:lle.

$A \longrightarrow M:$	$E_M(N_A, A)$	(1)
$M(A) \longrightarrow B:$	$E_B(N_A, A)$	(1')
$B \longrightarrow M(A):$	$E_A(N_A, N_B)$	(2')
$M \longrightarrow A:$	$E_A(N_A, N_B)$	(2)
$A \longrightarrow M:$	$E_M(N_B)$	(3)
$M \longrightarrow B:$	$E_B(N_B)$	(3')

Bob luulee olevansa yhteydessä Alicen kanssa, mutta onkin yhteydessä Malloryn kanssa. Mallory myös tuntee nonssit ja niistä generoitavan istuntoavaimen.

Turvallisuustavoitteet

- **Salaisten** avainten on pysyttävä salaisena.

  Istuntoavaimen pitää olla **tuore**.

  Joko luotettu osapuoli määrää avaimen, toinen osapuoli määrää sen yksin tai **molemmat vaikuttavat avaimen arvoon**.

- **Autentikointi** on aina kiinteä osa avaintenjakelua. Sekä julkisten että salaisten avainten pitää olla aitoja. Kenen kanssa yhteinen istuntoavain luodaan?

  Autentikointi voi olla **yksisuuntainen tai kaksisuuntainen** (= keskinäinen) (*unilateral, mutual*) .

  Kuinka yhdistetään autentikointi ja avainten jakelu?

- **Jakelun onnistumisesta** voidaan varmistua. (*key confirmation*)

  Implisiittisessä autentikoinnissa tunnistus tapahtuu vasta avainta käytettäessä, **eksplisiittisessä** protokollan aikana.

- **Kiistämättömyys** on tarpeen ainakin, jos yhteydestä laskutetaan.

Avaintenjakelun toimintaympäristö

- Kryptografiset algoritmit oletetaan luotettaviksi.
- Hyökkääjä (*adversary, intruder, opponent, enemy, attacker, eavesdopper, impersonator*) yrittää
  - saada istuntoavaimen selville
  - vaihtaa istuntoavaimen omakseen tai vanhaksi
  - esiintyä väärällä nimellä
  - estää muuten turvallisuustavoitteiden saavuttamisen
  - käyttää avaintenvaihtoprotokollaa hyväkseen muissa hyökkäyksissä.
- Hyökkääjä voi olla myös laillinen protokollaosapuoli tai useampien **salaliitto**.
- Hyökkäys on **passiivinen** tai **aktiivinen**.

  Passiivinen hyökkääjä osaa kuunnella kommunikaatiokanavia.

  Aktiivinen hyökkääjä osaa manipuloida (poistaa, muuttaa, lisätä) sanomia.

- **Oletetaan hyökkääjän olevan laaja salaliitto, joka pystyy vapaasti manipuloimaan liikennettä kaikissa epäluotettavissa kanavissa.**

Avaintenhallinnan kustannukset

- **Yhteyksien määrä** eri osapuolien välillä.
- Lähetettyjen **sanomien määrä** (*round-trips*)
- Kommunikaation **kaistanleveys** eri kanavissa, sanomien bittien määrä
- **Laskennan vaativuus** asiakkaan, palvelimen ja luotetun osapuolen päätelaitteissa
- **On-line** -laskenta
- Satunnaislukujen generointi
- **Luotettu osapuoli** *on-line / off-line*

  (*trusted third party TTP, authentication server, key distribution center KDC, key translation center KTC, certification authority CA*)
- Tarkojen **kellojen** tarve
- Kustannuksiin vaikuttaa merkittävästi, käytetäänkö salaisen vai julkisen avaimen algoritmeja.

Hyökkäykset

- Useimmat hyökkäykset ovat **toistohyökkäyksiä**.

  Toistohyökkäyksessä hyökkääjä nauhoittaa sanoman ja kopioi sen **protokollaan, sanomaan, sanoman osaan tai protokolla-ajoon**, jonne se ei kuulu.

  Toistohyökkäykset estetään sitomalla sanoma sen oikeaan yhteyteen.
- Hyökkäystyyppejä:
  - tunnistusongelma
  - tuoreusongelma (*freshness problem*)
  - *man-in-the-middle*
  - oraakkelihyökkäys (*oracle attack*)
  - lomitushyökkäys (*interleaving attack*) .

Tunnistettavuusongelma

Aidon sanoman erottaminen satunnaisesta

- Mikä tahansa bittijono näyttää salakieleltä. Sillä on jokin (satunnainen) selväkielinen vastine.
- Istuntoavain K_{SES} on satunnainen. Avainta katsomalla ei voi päätellä, onko avain aito vai ei.

  Bob ei pysty erottamaan aitoa sanomaa $E_{AB}(K_{SES})$ väärästä.
- Hyökkäys: Mallory lähettää Bobille satunnaisen bittijonon. Bob luulee saaneensa uuden avaimen. Mallory ei tiedä avainta, mutta pystyy näin häiritsemään Bobia.
- Lisätään selväkieleen vähintään 64 bittiä redundanssia, jotta Bob **tunnistaa** oikeat ja väärät sanomat.

  Lisätään otsikkotietoja tai maaginen numero. (Salauksen oletetaan suojaavan myös sanoman eheyttä.)

  $$1. \quad A \longrightarrow B: \quad A, E_{AB}(\text{"uusi avain"}, K_{SES})$$

  Nyt Bob erottaa aidot ja väärät sanomat. Mallory ei pysty luomaan oikean näköistä sanomaa.
- Valitettavasti tunnistettavuus huomioidaan usein vasta toteutuksessa.

Klassinen toistohyökkäys

(*classic replay*)

- Vaikka vastustaja ei pysty väärentämään tai muuttamaan salattu tekstia, hän voi nauhoittaa sanoman.

  $$A \longrightarrow B: \quad A, E_{AB}(K_{SES})$$

  Mallory näkee sanoman ja tallettaa sen. Myöhemmin Alice tai Bob käy huolettomaksi ja paljastaa vanhan avaimen K_{SES}.

  Vanhojen istuntoavaimien oletetaan aina paljastuvan.

  $$M(A) \longrightarrow B: \quad A, E_{AB}(K_{SES}) \qquad \text{Toistohyökkäys!}$$

  Bob luulee, että K_{SES} on hänen ja Alicen välinen uusi avain.
- Bob voi pitää kirjaa vanhoista avaimista/viesteistä, mutta se on raskasta.

Tuoreus

- Sanoma tai avain on **tuore**, jos sitä ei ole kopioitu aiemmista protokolla-ajoista.
- Ratkaisu 1: **aikaleima**

  Liitetään sanomaan lähetysaika. Hyväksytään vain uudet sanomat.

  $$1. \quad A \longrightarrow B: \quad A, E_{AB}(T_A, \ldots, K_{SES})$$

  Ongelma 1: Mikä on riittävän tuore?

  Ongelma 2: Kellojen oltava luotettavia. Hajautetussa järjestelmässä ei ole tarkkaa globaalia aikaa.
- Ratkaisu 2: **nonssi**

  Sanoman osana pitää olla jotain uutta tietoa, jota vanhoissa sanomissa ei ole. N_A on Alicen generoima nonssi.

  $$1. \quad A \longrightarrow B: \quad E_{AB}(T_A, N_A, \ldots, K_{SES})$$

  Bob säilyttää nonssit aikaleiman voimassaoloajan ja tarkistaa, että samaa ei vastaanoteta kahdesti.

  $$\begin{array}{lll} 1. & B \longrightarrow A: & B, N_B \\ 2. & A \longrightarrow B: & E_{AB}(N_B, \ldots, K_{SES}) \end{array}$$

  Bob tietää, että sanoma on tuore, koska se sisältää hänen generoimansa tuoreen nonssin. Ei tarvitse luottaa kelloon.

Oraakkelihyökkäys

- **Oraakkeli** antaa vastauksen vaikeaan ongelmaan selittämättä, miten ratkaisu löytyi.
- Haaste–vaste eräästä protokollasta:

  $$\begin{array}{lllll} 1. & A \longrightarrow B: & N_A & & \text{haaste} \\ 2. & B \longrightarrow A: & E_{SES}(N_A) & & \text{vaste} \end{array}$$

  Bob toimii oraakkelina, joka salaa minkä tahansa Malloryn lähettämän sanoman.

  Bob voi forwardoida autentikoinnin tai käyttää salattuja sanomia muihin tarkoituksiin.

  $$\begin{array}{lll} A \longrightarrow M(B): & N_A & \text{haaste} \\ M(A) \longrightarrow B: & N_A & \\ B \longrightarrow M(A): & E_{SES}(N_A) & \text{vaste} \\ M(B) \longrightarrow A: & E_{SES}(N_A) & \end{array}$$

Eksplisiittisyys

- Useimmat oraakkeli- ja lomitushyökkäykset johtuvat sanomien pituuden liiasta optimoinnista.

  Sanomiin lisää redundanttia tietoa!
- **Eksplisiittinen tyypitys**:

  Merkitään kaikkiin allekirjoitettuihin ja salattuihin sanomiin ja sanoman osiin **protokollan nimi, sanoman numero, sanoman osa tai primitiivinen tietotyyppi**.
- **Maksimaalinen informaatio** (*full information principle*) :

  Liitetään kaikkiin sanomiin kaikki protokolla-ajoon liittyvä tieto, josta lähettäjän ja vastaanottajan pitäisi olla yhtä mieltä.
- Esim. muunnettu X.509:

$$\begin{array}{lll} 1. & A \longrightarrow B & T_A, N_A, E_B(K_{AB}), \\ & & S_A(\text{“Msg 1 of my protocol”}, \\ & & T_A, A, B, K_A^+, K_B^+, N_A, K_{AB}) \\ 2. & B \longrightarrow A & T_B, N_B, N_A, E'_A(K_{BA}), \\ & & S_B(\text{“Msg 2 of my protocol”}, \\ & & T_A, T_B, A, B, K_A^+, K_B^+, N_A, N_B, K_{AB}, K_{BA}) \\ 3. & A \longrightarrow B & S_A(\text{“Msg 3 of my protocol”}, \\ & & T_A, T_B, A, B, K_A^+, K_B^+, N_A, N_B, K_{AB}, K_{BA}) \end{array}$$