AUTENTIMINE JA VÕTMETE LEVITAMINE

Needham-Schroederi algoritm (salajased võtmed)

Subjektide nimed X ja nende salajased võtmed KX:

U, KU - jaam U
V, KV - jaam V
A - autentimise server (teab subjektide nimesid ja ja nende võtmeid)

Tähistus: sõnumi M krüpteerimine võtmega KX

KX { M }

Jaam U tahab turvaliselt suhelda jaamaga V

1. Jaam U palub serverilt A võtit suhtlemiseks jaamaga V:

   U => A { U, V, n }

   n – juhuslik number (*a nonce*)

2. Server A saadab jaamale U võtme ja "pileti" suhtlemiseks jaamaga V:

   A => U KU { n, V, KUV, KV { KUV, U } }

   KUV – seansi U ⟷ V salajane võti
   { KUV, U } – pilet (*ticket*)

3. Jaam U saadab pileti jaamale V:

   U => V KV { KUV, U }

4. Jaam V saadab jaamale U “proovipalli”:

   V => U KUV { m }

   m – juhuslik arv

5. Jaam U viskab palli jaamale V tagasi:

   U => V KUV { m-1 }

NB! Ükski salajane võti ei liigu võrgus lahtisel kujul!

Algoritmi puudus – ei arvesta piletite aegumist

KERBEROS
(M.I.T. 1988)

Subjektide nimed X ja salajased võtmed KX:

C, KC	- klient
S, KS	– (andme)server
A	- autentimise server
T, KT	– piletite jagamise server

Eeldus: serverid A ja T asuvad ühes jaamas

<u>Klient C soovib suhelda serveriga S</u>

1. Klient C saadab autentimise serverile A palve anda luba suhelda piletite jagamise serveriga T:

   C => A { C, T, n } // lahtine sõnum!

   n – juhuslik arv

2. Autentimise server A annab kliendile C seansipileti suhtlemiseks serveriga T:

   A => C KC { KCT, n }, KT { pilet(C, T) }

   KC – kliendi salajane võti
   KCT – seansi C ⟷ T võti
   KT – serveri T salajane võti

   pilet(C, Z) = { C, Z, t1, t2, KCZ }

   KCZ – sessiooni C ⟷ Z võti
   [t1, t2] – pileti kehtivusaeg

3. Klient C küsib serverilt T piletit suhtlemiseks serveriga S:

   C => T KCT { C, t }, KT { pilet(C, T) }, S, n

   t – ajatempel (*time-stamp*)

4. Server T annab kliendile C seansipileti suhtlemiseks serveriga S:

   T => C KCT { KCS, n }, KS { pilet(C, S) }

   KCS - seansi C ⟷ T võti
   KS – serveri S salajane võti

5. Klient C saadab päringu serverile S:

   C => S KCS { C, t }, KS { pilet(C, S) }, päring, n

   päring – vajadusel krüpteeritud võtmega KCS

6. Server S vastab kliendile C:

   S => C vastus, KCS { n }

   vastus – vajadusel krüpteeritud võtmega KCS

DIGITAALNE ALLKIRI

Needham-Schroederi algoritm (salajane võti)

Subjektide nimed X ja nende salajased võtmed KX:

U, KU - kasutaja U
V, KV - kasutaja V
A - autentimise server (teab subjektide nimesid ja ja nende võtmeid)

U saadab V-le sõnumi M ja digitaalse allkirja

1. Kasutaja U saadab sõnumi M krüptograafilise lühendi (*digest*) serverile A

   U =>A { U, KU { hash(M) } }

   hash( ) - räsifunktsioon

2. Server A moodustab kasutaja U sertifikaadi ja saadab selle U-le

   A => U KA { U, hash(M), t }

   t – ajatempel

3. Kasutaja U saadab V-le sõnumi ja vastava sertifikaadi

   U => V M, KA { U, hash(M), t }

4. Kasutaja V saadab sertifikaadi serverile A

   V => A V, KA { U, hash(M), t }

5. Server A kontrollib sertifikaati ja saadab selle V-le

   A => V KV { U, hash(M), t }

6. Kasutaja U kontrollib ka sõnumi terviklikkust, arvutades uuesti sõnumi krüptograafilise lühendi ja võrreldes seda sertifikaadis olevaga.