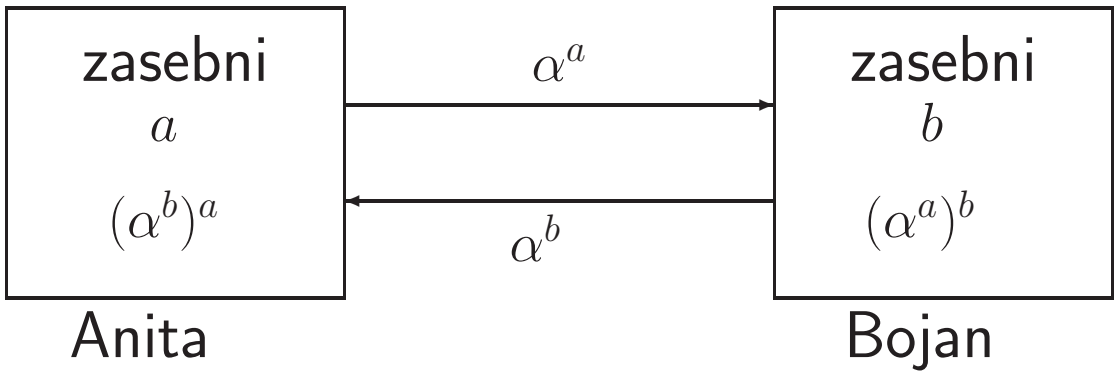
Diffie-Hellmanova uskladitev ključev

Naj bo p praštevilo in α generator multiplikativne grupe $\mathbb{Z}_p^*$. Naj bosta oba javno poznana (ali pa naj ju oseba U sporoči osebi V).

1. *Oseba U izbere naključen a_U, $0 \leq a_U \leq p-2$, izračuna $\alpha^{a_U} \bmod p$ in ga pošlje osebi V.*
2. *Oseba V izbere naključen a_V, $0 \leq a_V \leq p-2$, izračuna $\alpha^{a_V} \bmod p$ in ga pošlje osebi U.*
3. *Osebi U in V izračunata zaporedoma $K = (\alpha^{a_V})^{a_U} \bmod p$ in $K = (\alpha^{a_U})^{a_V} \bmod p$.*



Anita in Bojan si delita skupni element grupe:

$$(\alpha^a)^b = (\alpha^b)^a = \alpha^{ab}.$$

Edina razlika med tem protokolom in pa Diffie-Hellmanovim protokolom za distribucijo ključev je, da si izberemo nova eksponenta a_U in a_V uporabnikov U in V zaporedoma vsakič, ko poženemo ta protokol.

Varnost Diffie-Hellmanovega protokola

Protokol ni varen pred aktivnim napadalcem,
ki prestreže sporočila in jih nadomesti s svojimi.
Ta napad bomo imenovali **napad srednjega moža**.

$$U \quad \underset{\alpha^{a'_V}}{\overset{\alpha^{a_U}}{\rightleftarrows}} \quad W \quad \underset{\alpha^{a_V}}{\overset{\alpha^{a'_U}}{\rightleftarrows}} \quad V$$

Na koncu sta osebi U in V vzpostavili z napadalcem W zaporedoma ključa $\alpha^{a_U a'_V}$ in $\alpha^{a'_U a_V}$.

Tako bo zašifrirano sporočilo osebe U odšifriral napadalec W ne pa oseba V.

Uporabnika U in V bi bila rada prepričana, da ni prišlo namesto medsebojne izmenjave sporočil do izmenjave z napadalcem W.

Potrebujeta protokol za medsebojno avtentikacijo (predstavitev).

Dobro bi bilo, če bi potekala avtentikacija istočasno z uskladitvijo ključev, saj bi s tem onemogočili aktivnega napadalca.

Overjena uskladitev ključev

Diffie, Van Oorschot in Wiener so predlagali protokol **uporabnik-uporabniku** (station-to-station - STS), ki je protokol za *overjeno uskladitev kjuča* in je modifikacija Diffie-Hellmanove uskladitve ključev.

Vsak uporabnik ima **certifikat (potrdilo)**

$$C(U) = \Big(\mathrm{ID}(U), \mathrm{ver}_U, \mathrm{sig}_{\mathrm{TA}}\big(\mathrm{ID}(U), \mathrm{ver}_U\big)\Big),$$

kjer je shranjena njegova identifikacija $\mathrm{ID}(U)$.

Poenostavljen protokol uporabnik-uporabniku

1. *Oseba U izbere naključen $a_U \in \{0, \ldots, p-2\}$, izračuna* $\alpha^{a_U} \bmod p$ *in pošlje osebi V.*

2. *Oseba V izbere naključen $a_V \in \{0, \ldots, p-2\}$, izračuna* $\alpha^{a_V} \bmod p$,
   $K = (\alpha^{a_U})^{a_V} \bmod p$ in $y_V = \mathrm{sig}_V(\alpha^{a_V}, \alpha^{a_U})$,
   ter pošlje potrdilo $(C(V), \alpha^{a_V}, y_V)$ *osebi U.*

3. *Oseba U izračuna* $K = (\alpha^{a_V})^{a_U} \bmod p$ *ter preveri podpis* y_V *z uporabo* ver_V *in potrdilo* $C(V)$ *z* $\mathrm{ver}_{\mathrm{TA}}$.

   Nato izračuna $y_U = \mathrm{sig}_U(\alpha^{a_U}, \alpha^{a_V})$ *in pošlje potrdilo* $(C(U), y_U)$ *osebi* V.

4. *Oseba V preveri podpis* y_U *z uporabo* ver_U *in potrdilo* $C(U)$ *z uporabo* $\mathrm{ver}_{\mathrm{TA}}$.

Varnost protokola STS

Uporabnika U in V si izmenjata naslednje informacije (izpustimo potrdila):

$$
\begin{array}{ccc}
 & \xrightarrow{\quad \alpha^{a_U} \quad} & \\
U & \xleftarrow{\quad \alpha^{a_V},\ \mathrm{sig}_V(\alpha^{a_V}, \alpha^{a_U}) \quad} & V \\
 & \xrightarrow{\quad \mathrm{sig}_U(\alpha^{a_U}, \alpha^{a_V}) \quad} &
\end{array}
$$

Kaj lahko naredi napadalec W (mož na sredini):

$$
\begin{array}{ccccc}
 & \xrightarrow{\alpha^{a_U}} & & \xrightarrow{\alpha^{a'_U}} & \\
U & \xleftarrow{\alpha^{a'_V},\ \mathrm{sig}_V(\alpha^{a'_V},\alpha^{a_U})=?} & W & \xleftarrow{\alpha^{a_V},\ \mathrm{sig}_V(\alpha^{a_V},\alpha^{a'_U})} & V \\
 & \xrightarrow{\mathrm{sig}_U(\alpha^{a_U},\alpha^{a'_V})} & & \xrightarrow{\mathrm{sig}_U(\alpha^{a'_U},\alpha^{a_V})=?} &
\end{array}
$$

Poenostavljeni STS protokol je torej varen pred napadom srednjega moža.

Tako oblikovan protokol ne vsebuje potrditve ključa, kakor je slučaj v Kerberosovi shemi.

Protokol, v katerem je vključena potrditev ključa:

$$y_V = e_K(\mathrm{sig}_V(\alpha^{a_V}, \alpha^{a_U})), \quad y_U = e_K(\mathrm{sig}_U(\alpha^{a_U}, \alpha^{a_V}))$$

se imenuje STS protokol.

MTI protokoli

Matsumoto, Takashima, Imai so modificirali Diffie-Hellmanovo uskladitev ključev, tako da uporabniki U in V ne potrebujejo podpisov.

Kadar moramo izmenjati dve pošiljki, pravimo, da gre za **protokole z dvema izmenjavama**.

Predstavili bomo en njihov protokol.

Osnovne predpostavke so enake kot pri Diffie-Hellmanovi uskladitvi ključev: praštevilo p in generator α multiplikativne grupe $\mathbb{Z}_p^*$ sta javna.

Vsak uporabnik U ima svoj *zasebni* eksponent a_U $(0 \leq a_U \leq p-2)$ in *javno* vrednost $b_U = \alpha^{a_U} \bmod p$.

Agencija TA ima shemo za digitalni podpis, z *javnim* algoritmom $\mathrm{ver}_{\mathrm{TA}}$ in *tajnim* algoritmom $\mathrm{sig}_{\mathrm{TA}}$.

Vsak uporabnik U ima svoj certifikat:

$$C(U) = (\mathrm{ID}(U), b_U, \mathrm{sig}_{\mathrm{TA}}(\mathrm{ID}(U), b_U)).$$

1. *Oseba U izbere naključen $r_U \in \{0, \ldots, p-2\}$, izračuna* $s_U = \alpha^{r_U} \bmod p$ *in pošlje osebi V* $(C(U), s_U)$.

2. *Oseba V izbere naključen $r_V \in \{0, \ldots, p-2\}$, izračuna* $s_V = \alpha^{r_V} \bmod p$ *in pošlje osebi U* $(C(V), s_V)$.

3. *Osebi U in V izračunata zaporedoma*

$$K = s_V^{a_U} b_V^{r_U} \bmod p \quad \text{in} \quad K = s_U^{a_V} b_U^{r_V} \bmod p,$$

kjer sta b_V in b_U zaporedoma iz $C(V)$ in $C(U)$.

Varnost protokola MTI

Ta MTI protokol je enako varen pred pasivnimi sovražniki kot Diffie-Hellmanov protokol.

Varnost pred aktivnimi sovražniki je bolj vprašljiva. Brez uporabe podpisnega algoritma nismo varni pred napadom srednjega moža.

$$U \underset{C(V),\ \alpha^{a_V} \bmod p}{\overset{C(U),\, \alpha^{r_U} \bmod p}{\rightleftarrows}} V$$

Ključ uporabnikov, ki komunicirata, je težko izračunati, ker je v ozadju težko izračunljiv diskretni logaritem.

Tej lastnosti pravimo **implicitna overitev ključev**.

Uskladitev ključev s ključi, ki se sami overijo

Giraultova shema ne potrebuje certifikatov, saj uporabnike razlikujejo že njihovi javni ključi in identifikacije.

Vsebuje lastnosti RSA sheme in diskretnega logaritma.

Uporabnik naj ima identifikacijo ID(U).
Javni ključ za osebno overitev dobi od agencije TA.

Naj bo $n = p\,q$, kjer je $p = 2p_1 + 1$, $q = 2q_1 + 1$, in so p, q, p_1, q_1 velika praštevila. Potem je

$$(\mathbb{Z}_n^*, \cdot) \sim (\mathbb{Z}_p^* \times \mathbb{Z}_q^*, \cdot).$$

Največji red poljubnega elementa v $\mathbb{Z}_n^*$ je najmanjši skupni večkratnik elementov $p - 1$ in $q - 1$ oziroma $2p_1q_1$.

Naj bo α generator ciklične podgrupe v $\mathbb{Z}_p^*$ reda $2p_1q_1$, problem diskretnega logaritma v tej podgrupi pa naj bo računsko prezahteven za napadalca.

Javni ključ za osebno overitev

Naj bosta števili n, α javni,
števila p, q, p_1, q_1 pa naj pozna samo agencija TA.

Število e je javni RSA šifrirni eksponent in ga izbere agancija TA, $d = e^{-1} \bmod \varphi(n)$ pa je tajni odšifrirni eksponent.

1. *Oseba U izbere tajni eksponent a_U,*
   izračuna $b_U = \alpha^{a_U} \bmod n$ in
   izroči a_U ter b_U agenciji TA.
2. *Agencija TA izračuna*
   $p_U = (b_U - \mathrm{ID}(U))^d \bmod n$ ter ga izroči osebi U.

Giraultov protokol za uskladitev ključev

1. *Oseba U izbere naključen zasebni r_U, izračuna*
$$s_U = \alpha^{r_U} \bmod n$$
ter pošlje $\mathrm{ID}(U)$, p_U *in* s_U *osebi* V.
2. *Oseba V izbere naključen zasebni r_V, izračuna*
$$s_V = \alpha^{r_V} \bmod n$$
ter pošlje $\mathrm{ID}(V)$, p_V *in* s_V *osebi* U.
3. *Osebi U in V izračunata ključ K zaporedoma z*

$$s_V^{a_U}(p_V^e + \mathrm{ID}(V))^{r_U} \bmod n, \quad s_U^{a_V}(p_U^e + \mathrm{ID}(U))^{r_V} \bmod n.$$

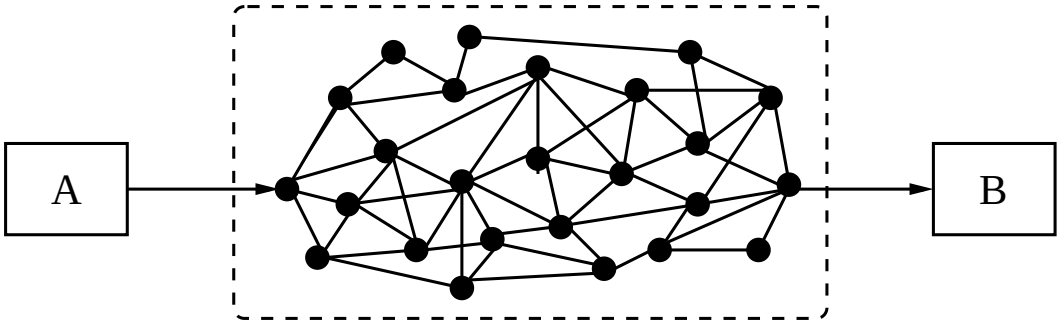
Varnost Giraultovega protokola

Ključ za osebno overitev varuje pred sovražniki.

Protokol implicitno overi ključe, zato napad srednjega moža ni možen.

Agencija TA je prepričana, da uporabnik pozna vrednost števila a predno izračuna ključ za osebno overitev.

Internetne aplikacije

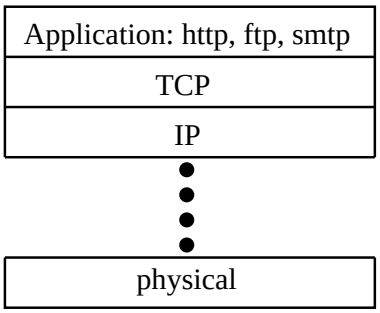


- ftp: File Transfer Protocol
- http: HyperText Transfer Protocol
- smtp: Simple Mail Transfer Protocol

TCP – Transport Control Protocol
IP – Internet Protocol

TCP/IP

Protokolov sklad:



TCP/IP paket:

IP header	TCP header	Application header	Data

IP header:	source address	destination address	------

Nekateri napadi

- **IP address spoofing** (slov. ponarejanje naslovov)
  rešitev: overi glavo IP paketa
- **IP packet sniffing** (slov. vohljanje za IP paketi)
  rešitev: zašifriraj IP payload (vse kar se prenaša)
- **Traffic analysis** (slov. Analiza prometa)
  rešitev: zašifriraj pošiljateljev in prejemnikov naslov

Varnost znotraj TCP/IP

Varnostni protokoli so prisotni na različnih nivojih TCP/IP sklada.

1. IP nivo: IPsec.
2. Transportni nivo: SSL/TLS.
3. Aplikacijski nivo: PGP, S/MIME, SET, itd.

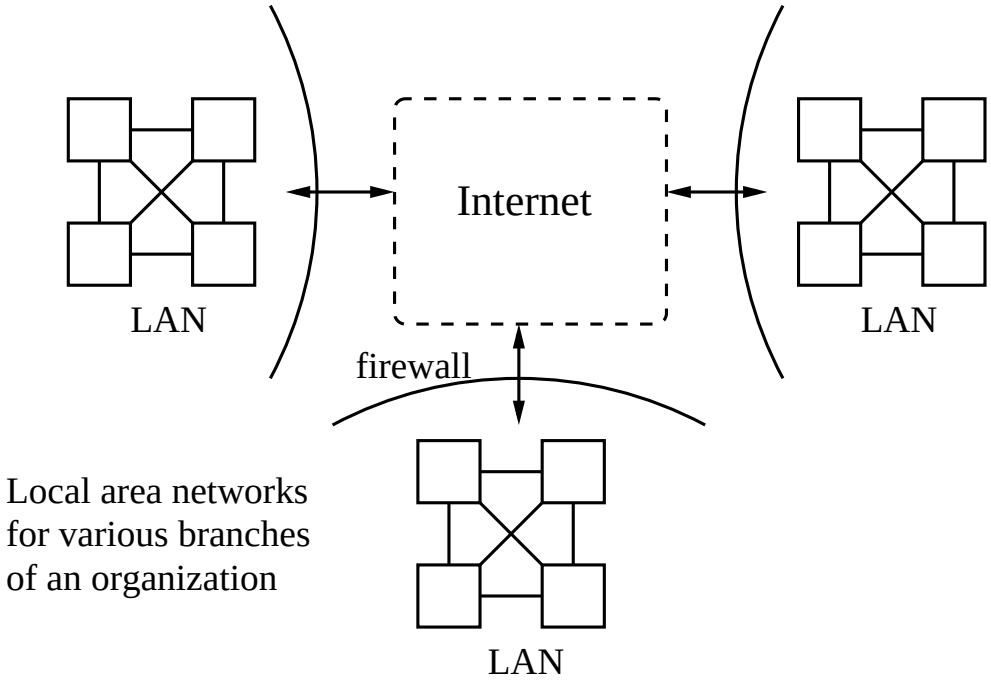
Internet Engineering Task Force (IETF)

- Sprejema standarde za razvoj Internetne arhitekture in omogoča nemoteno delovanje Interneta.
- Odprta za vse zainteresirane posameznike: `www.ietf.org`
- Delo, ki ga opravljajo delovne skupine povezane z varnostjo (Security Area) pokrivajo:

- IP Security Protocol (IPsec)
- Transport Layer Security (TLS)
- S/MIME Mail Security
- Odprto specifikacijo za PGP (OpenPGP)
- Secure Shell (secsh)
  (Nova verzija ssh protokola, ki omogoča varno prijavo na oddaljene šifre in varen prenos datotek.)
- X.509 Public-Key Infrastructure (PKIX)

IPsec: Virtual Private Networks (VPNs)

Omogočajo šifriranje in overjanje (overjanje izvora podatkov, celovitost podatkov) na IP layer.



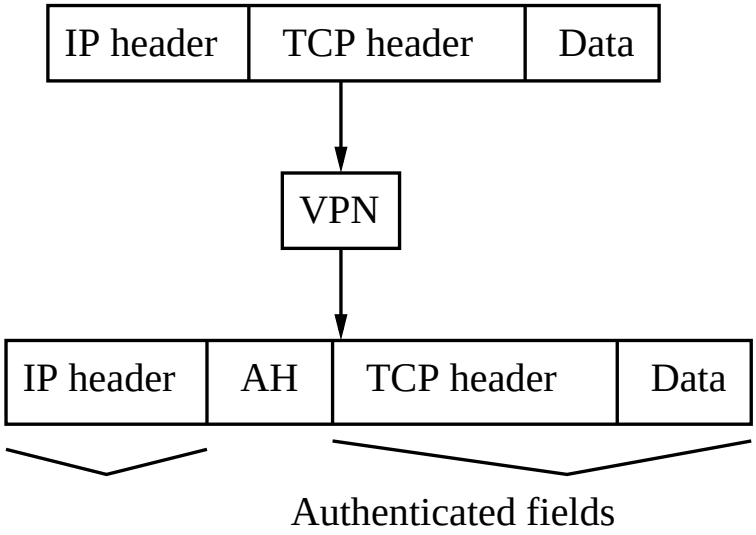
Gradniki IPsec

- Security Association (SA):
  - upravlja algoritme in ključe med sogovorniki,
  - vsaka glava IPsec se nanaša na Security Association preko Security Parameter Index (SPI).
- Upravljanje s ključi:
  - dogovor o ključu z Diffie-Hellmanovo shemo (OAKLEY),
  - kreira ključe za Security Association,
  - upravljanje z javnimi ključi, ki ni pokrito v IPsec.
- Trije načini IPsec servisov:
  - AH: overjanje,
  - ESP: šifriranje + overjanje.

IPsec glava za overjanje (AH)

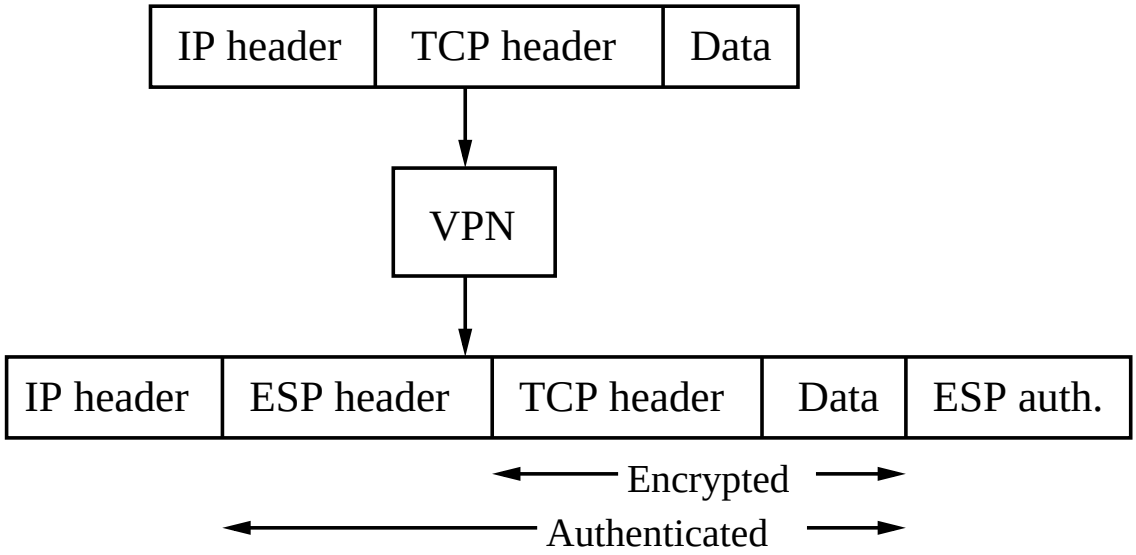
– Podpira MACs: HMAC-MD5-96, HMAC-SHA-1-96.

– Transportni način:



IPec ESP glava

- Encapsulating Security Payload.
- Podprti šifrirni algoritmi: 3-DES, RC5, IDEA, ...
- Transportni način:

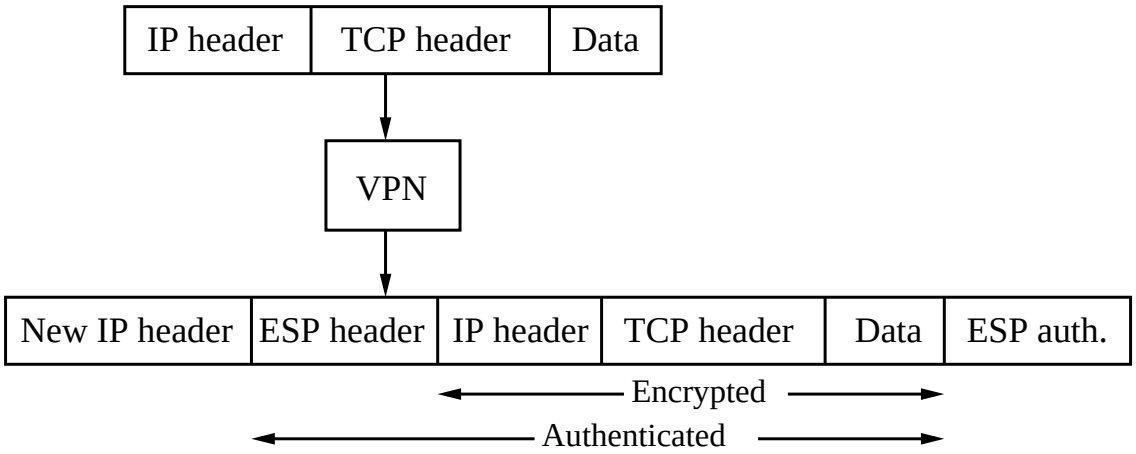


- Opomba: analiza prometa je še vedno možna (ker IP glave niso šifrirane).

ESP v tunelskem načinu

– Požarni zid vključi novo IP glavo
(IP naslov pošiljateljevega požarnega zidu in
IP naslov prejemnikovega požarnega zidu).

– Možna je samo zelo omejena analiza prometa.



Secure Sockets Layer (SSL)

- SSL je naredil Netscape.
- TLS (Transport Layer Security) je IETF-ova verzija SSL-a.
- SSL uporabljamo v brskalnikih (npr. Netscape) za zaščito mrežnih transakcij.
- Osnovne komponente SSL/TLS:

  handshake protocol: dopusti strežniku in klientu, da se overita in dogovorita za kriptografske ključe,

  record protocol: uporabljan za šifriranje in overjanje prenašanih podatkov.

Upravljanje z javnimi ključi v SSL/TLS

- Korenski CA ključ je vnaprej inštaliran v brskalnik.
  - Klik na "Security" in nato na "Signers", da najdete seznam ključev korenskih CA v Netscape-u.
- Mrežnim strežnikom certificirajo javne ključe z enim izmed korenskih CA-jev (seveda brezplačno).
  - Verisign-ov certification business za mrežne strežnike
    `www.verisign.com/server/index.html`

- Klienti (uporabniki) lahko pridobijo svoje certifikate. Večina uporabnikov trenutno nima svojih lastnih certifikatov.
  - Če klienti nimajo svojih certifikatov, potem je overjanje samo enostransko (strežnik se avtenticira klientu).
  - Obiščite varno internetno stran kot npr. webbroker1.tdwaterhouse.ca in kliknite na "padlock" v Netscapu, da si ogledate informacijo o strežnikovem certifikatu.

SSL/TLS handshake protocol

Na voljo so naslednji kriptografski algoritmi:

- MAC: HMAC-SHA-1, HMAC-MD5.
- šifriranje s simetričnimi ključi: IDEA, RC2-40, DES-40, DES, Triple-DES, RC4-40, RC4-128.
- Osnovne sheme za dogovor o ključu so:

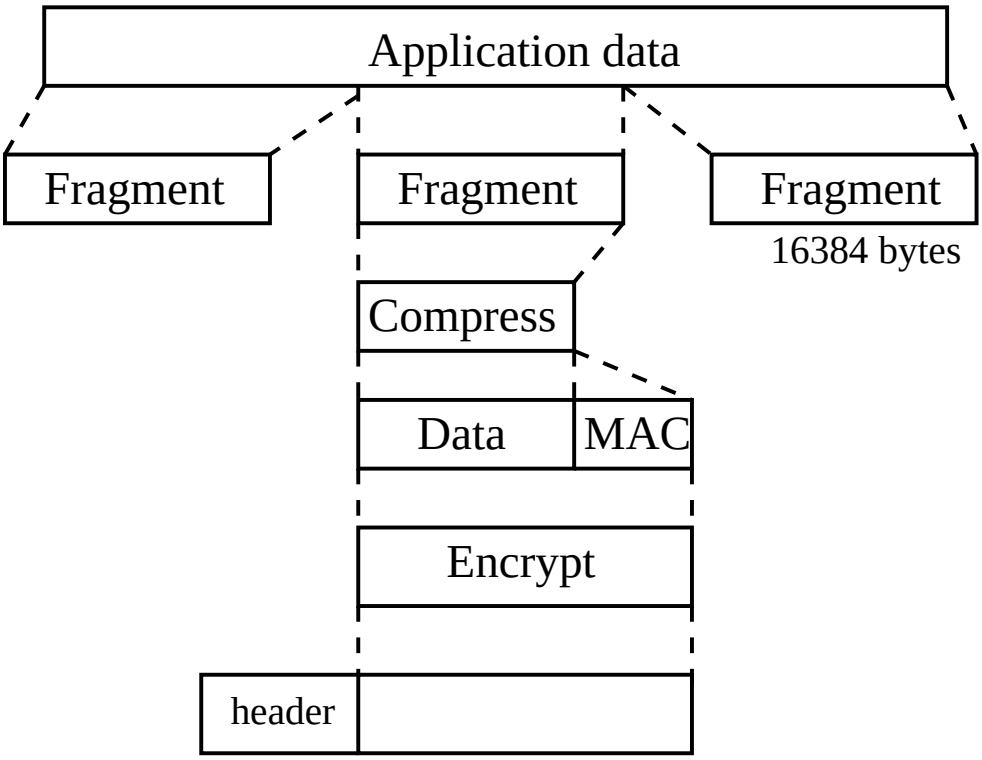
- RSA transport ključev: deljeno skrivnost izbere klient in jo zašifrirana s strežnikovim javnim RSA ključem.
- Fixed Diffie-Hellman: strežnikov Diffie-Hellman-ov javni ključ g^x je v njegovem certifikatu.
  Klient ima lahko g^y v svojem certifikatu, ali generira enkratno vrednost g^y.
- Ephemeral Diffie-Hellman: Strežnik izbere enkratni Diffie-Hellman-ov javni ključ g^x in ga podpiše s svojim RSA ali DSA ključem za podpise. Klient izbere enkratni g^y in ga podpiše če in samo če ima certifikat.
- MAC in šifrirni ključi so izpeljani iz skupne skrivnosti.

SSL/TLS handshake protokol (2)

1. faza: Določi varnostne zmožnosti.
   - Verzija protokola, način kompresije, kriptografski algoritmi,...
2. faza: Strežnikovo overjanje in izmenjava ključev.
   - Strežnik pošlje svoj certifikate, in (morda še) parametre za izmenjavo ključev.
3. faza: Klientovo overjanje in izmenjava ključeve.
   - Klient pošlje svoj certifikat (če ga ima) in parametre za izmenjavo ključev.
4. faza: Zaključek.

SSL/TLS record protocol

Predpostavimo, da klient in strežnik delita MAC tajnega ključa in sejni šifrirni ključ:



9. poglavje

Identifikacijske sheme

oziroma **sheme za predstavljanje**:

- Uporaba in cilji identifikacijskih shem
- Protokol z izzivom in odgovorom
- Schnorrova identifikacijska shema
- Okomotova identifikacijska shema
- Guillou-Quisquater identifikacijska shema
- Pretvarjanje identifikacijske sheme v shemo za digitalni podpis

Pogosto hočemo dokazati svojo identiteto, npr.:

- **dvig denarja**
  (na bankomatu rabimo kartico in PIN)
- **nakup/plačilo**
  (prek telefona, potrebujemo kartico in rok veljave)
- **telefonska kartica** (telefonska številka in PIN)
- **prijava na svojo šifro na računalniku**
  (uporabniško ime in geslo)

Cilji identifikacijskih shem

- priča Anitine predstavitve Bojanu se ne more kasneje lažno predstaviti za Anito,
- tudi Bojan se ne more po Anitini predstavitvi lažno predstaviti za Anito,
- enostavnost (npr. za pametno/čip kartico)

Anita s svojo predstavitvijo ne izda informacije, ki jo identificira/predstavlja.

Kartica se predstavi sama, nepooblaščeno uporabo (kraja/izguba) pa preprečimo s PIN-om.

Protokol z **izzivom in odgovorom**:

Anita in Bojan delita tajni (skrivni) ključ K, ki ga uporabljata za šifriranje.

1. *Bojan izbere 64-bitni izziv x in ga pošlje Aniti.*
2. *Anita izračuna $y = e_K(x)$ in ga pošlje Bojanu,*
3. *Bojan izračuna $y' = e_K(x)$ in preveri $y = y'$.*

Skoraj vse sheme uporabljajo protokole z izzivom in odgovorom, vendar pa najbolj koristne ne uporabljajo skupnih ključev.