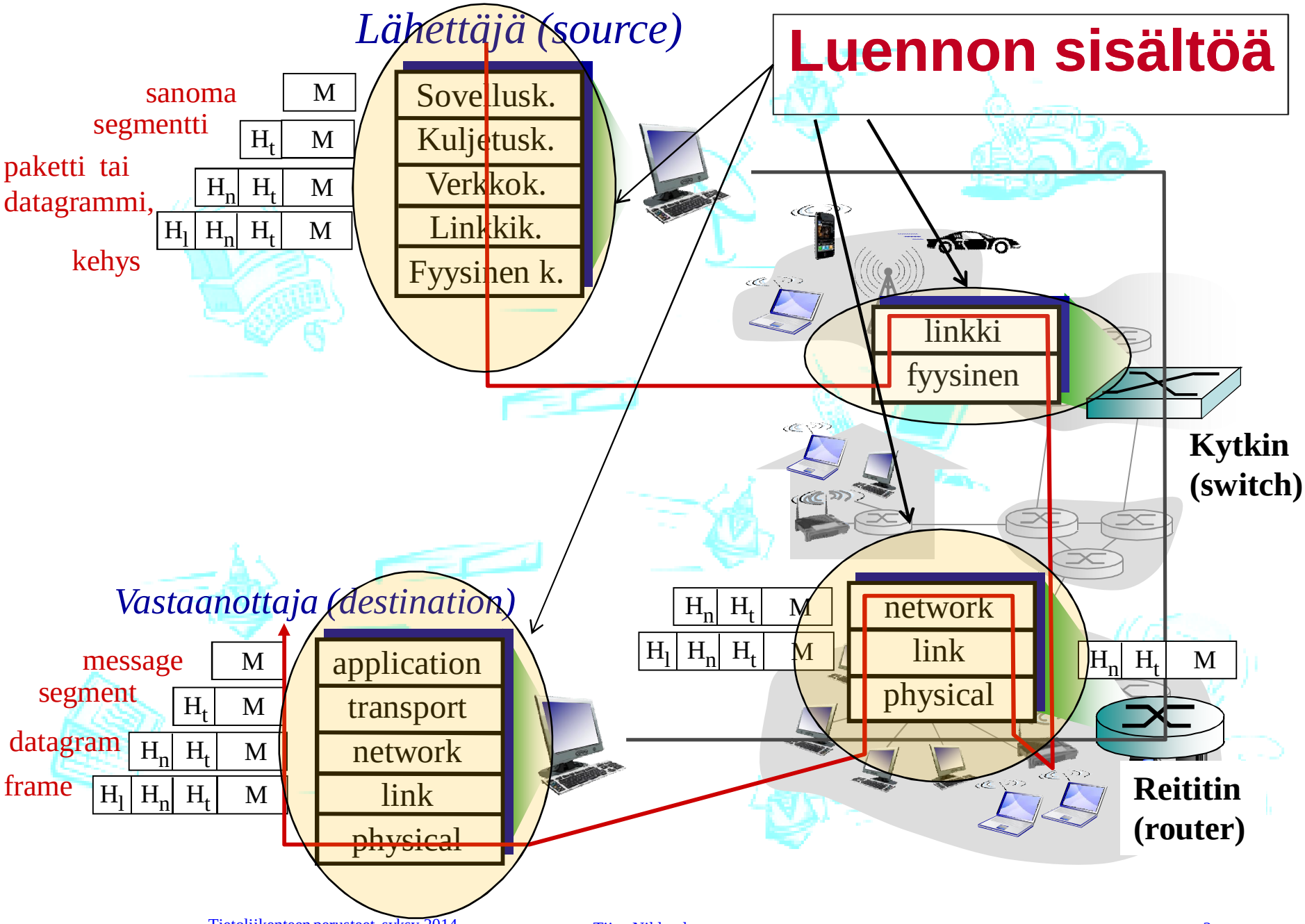
Tietoliikenteen perusteet

Luento 12: Tietoliikenteen turvallisuus: protokollat (kuten SSL, VPN, IPsec, WEP)

Syksy 2014, Tiina Niklander

Kurose&Ross: Ch 8

Luennon sisältöä



Sisältö

- **Tietoturvaan liittyviä protokollia eri kerroksilla:**

- **Kuljetuskerros: SSL, TLS**
- **Verkkokerros: IPsec, VPN**
- **Linkkikerros: WEP (vaikka ei olekaan enää turvallinen valinta) – luento 13**

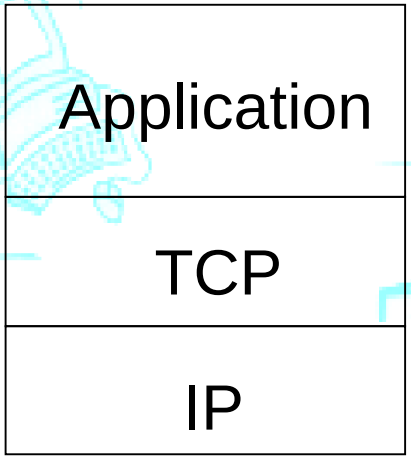
Oppimistavoitteet:
- Ymmärtää protokollien tietoturvan periaatteet
-Osaa kuvata tietoturvan kannalta keskeisten protokollien tavoitteet ja toimintaperiaatteet

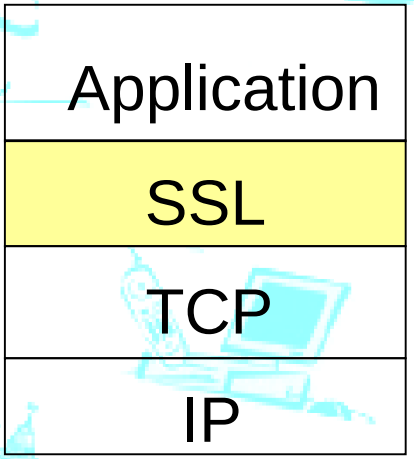
KULJETUSKERROS: SECURE SOCKETS LAYER SSL

SSL: Secure Sockets Layer

- Yleisesti käytetty suojausprotokolla
  - Selainten ja www-palvelujen käyttämä
  - http**s**
- mekanismi: [Woo 1994], toteutus: Netscape
- Uudemmat versiot = TLS: transport layer security, RFC 2246
- tarjoaa
  - *todentamisen, yksityisyyden, luottamuksellisuuden*
- Alkuperäisiä tavoitteita:
  - Verkkokauppojen transaktiot
  - salaus (esim. luottokortin numero)
  - www-palvelun todennus
  - Valinnainen käyttäjän todennus
  - Vähän työtä uusien palvelujen käyttöönotossa
- Käytettävissä kaikilla TCP-yhteyksillä
  - oma API: secure socket interface

SSL ja TCP/IP





Oikeastaan ISO/OSIn esitystapa-kerros

Sovellus ilman SSL:ää

Sovellus käyttää SSL:ää

- SSL tarjoaa sovellukselle suojatun kuljetuspalvelun. TCP:n kannalta se on sovelluskerroksella.
- Oma ohjelmointirajapinta (API) sovelluksille (korvaa TCP:n pistokerajapinnan, mutta samankaltainen)
- C ja Java SSL kirjastoja/luokkia yleisesti saatavilla

SSL:n toiminnallisuus

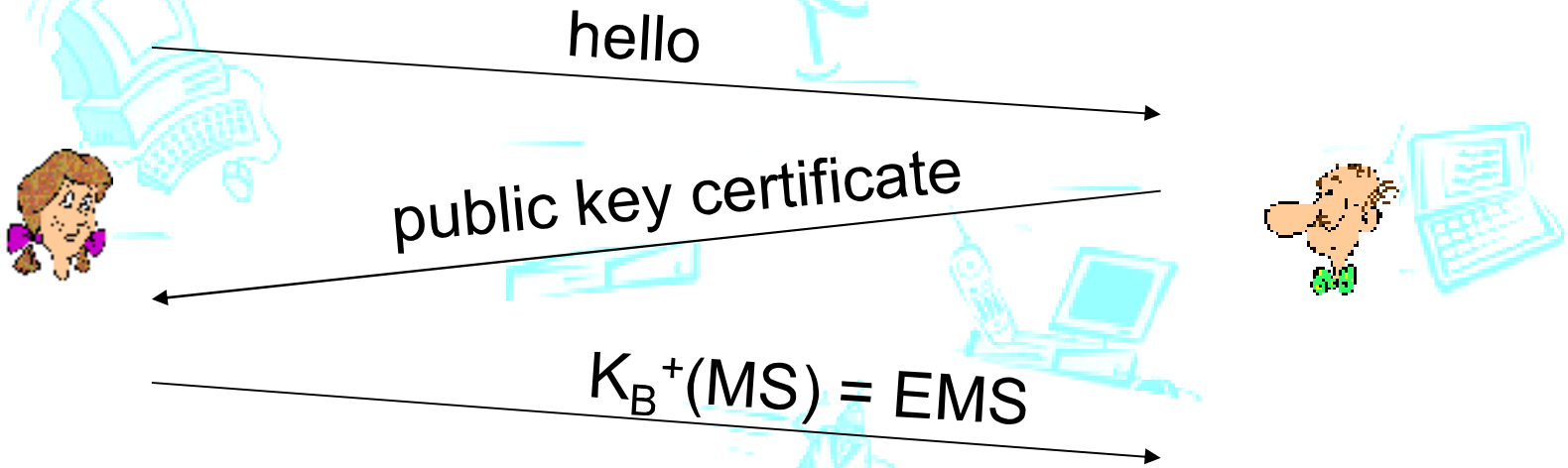
- Voisi toimia kuten Pretty Good Privacy (PGP) sähköpostin kanssa, mutta SSL käsittelee tavuvuota (byte stream) ja interaktiivista dataa
  - Suojattava paloittain, ei voi suojata ja tarkistaa vasta lopuksi
- SSL suojaa koko yhteyden alusta loppuun käyttäen avaimia
- Vastapuolen todentamiseksi avainten varmenteet (sertifikaatit) tarkastetaan kättelyvaiheessa

Yksinkertaistettu leluversio: Toy SSL

SSL:n perusidea, mutta yksinkertaistettuna

- *Kättely (handshake):* Alice ja Bob käyttävät varmennettuja julkisia (ja vastaavia salaisia) avaimia sekä sopivat yhteyden aikana käytettävästä jaetusta salaisuudesta (shared secret).
- *Avainten muodostaminen (key derivation):* Alice ja Bob muodostavat joukon avaimia jaetun salaisuuden perusteella
- *Tiedonsiirto (data transfer):* Tavuvirta pätkitään jonoksi tietueita (record), jotka suojataan erikseen
- *Yhteyden sulkeminen (connection closure):* Erikoisviestit, joilla yhteys suljetaan turvallisesti

Toy: yksinkertainen kättely



MS: pääsalaisuus (master secret), jaettu

EMS: salattu pääsalaisuus

Toy: avainten muodostaminen

- Jokaiselle kryptografiselle operaatiolle oma avain
  - Suojaus heikkenee, jos samaa avainta käytetään moneen tarkoitukseen
  - SIIS: eri avaimet viestien todennusosiolle (message authentication code, MAC) ja salaukselle
- Neljä avainta:
  - K_c = datan salausavain datalle asiakkaalta palvelijalle
  - M_c = todennusavain (MAC key) asiakkaan datalle
  - K_s = datan salausavain datalle palvelijalta asiakkaalle
  - M_s = todennusavain (MAC key) palvelijan datalle
- Avainten muodostusfunktio (key derivation function, KDF)
  - Avaimet muodostetaan pääsalaisuudesta (ja mahdollisesta satunnaisesta datasta)

Toy: tietueet

- Miksi TCP:n tavuvirta pitää pätkiä tietueiksi eikä suojata sellaisenaan?
  - Mihin tavuvirrassa voi sijoittaa todennusosion MAC? Jos vasta lopussa, tavuvirran eheyttä ei voi tarkistaa aiemmin.
  - Esim. Verkkokaupassa on tarve tarkistaa viestien eheys jo kesken kommunikoinnin eikä vasta lopuksi
- Tavuvirta tietueiksi
  - Kullakin tietueella oma todennusosio MAC
  - Vastaanottaja voi tarkistaa kunkin tietueen heti sen saavuttua
- HUOM: vastaanottajan pitää voida erottaa tietueesta todennusosio ja data
  - Halutaan kuitenkin käyttää vaihtelevanpituista tietuetta

Toy: järjestysnumero tietueille

- *Ongelma:* ilman järjestysnumeroa hyökkääjä voi tallettaa viestit ja toistaa ne myöhemmin tai vaihtaa viestien järjestyksen
- *Ratkaisu:* järjestysnumero mukaan todennusosioon MAC:
  - MAC = MAC(M_x, sequence||data)
  - Huom: ei erillistä kenttää järjestysnumerolle. Sitä ei lähetetä viestin osana, vaan käytetään vain todennusosion muodostamisessa

- *Ongelma:* Hyökkääjä voi silti toistaa koko viestiketjun
- *Ratkaisu:* Käytä haastetta (nonce) kättelyssä

Toy: yhteyden hallinta

- *Ongelma:* Typistyshyökkäys (truncation attack):
  - Hyökkääjä väärentää TCP yhteyden päättävän FIN-viestin
  - Todellisia viestejä jää välittymättä tämän johdosta.
- *Ratkaisu:* tietueille tyypit, joista yksi sulkemiselle
  - tyyppi 0 datalle; tyyppi 1 yhteyden päättämiselle
- MAC = MAC(M_x, sequence||type||data)
  - Todennusosioon mukaan myös tyyppi väärentämisen estämiseksi.

Toy SSL: yhteenveto

bob.com

hello

certificate, nonce

$K_B^+(MS) = EMS$

type 0, seq 1, data

type 0, seq 2, data

type 0, seq 1, data

salattuna

type 0, seq 3, data

type 1, seq 4, close

type 1, seq 2, close

Toy SSL:n puuttuvat tiedot

- Kuinka pitkiä kentät ovat?
- Mitä salausalgoritmia ja/tai –protokollaa käytetään?
- Miten neuvotellaan algoritmista?
  - Asiakas ja palvelija voivat käyttää ja tarjota useita erilaisia salausalgoritmeja
  - Asiakas ja palvelija voivat yhdessä sopia käytettävästä algoritmista ennen varsinaista tiedon siirtoa

SSL: salausmetelmien joukko

- Käytetyt salausmenetelmät
  - Julkisen avaimen salaus
  - Symmetrinen salaus
  - MAC salausalgoritmi
- SSL tukee erilaisia menetelmiä
- Neuvottelussa asiakas ja palvelin sopivat käytettävät menetelmät
  - Asiakas tarjoaa vaihtoehdot
  - Palvelin valitsee niistä yhden

Yleisiä SSL:n symmetrisiä salausmenetelmiä

- DES – Data Encryption Standard: lohkosalaus
- 3DES – 3-kert. DES: lohko
- RC2 – Rivest Cipher 2: lohko
- RC4 – Rivest Cipher 4: vuosalaus

SSL julkisen avaimen salaus

- RSA

Oikea SSL: kättely (1)

Tavoitteet

1. Palvelimen todentaminen
2. Neuvottelu: sovitaan salausmenetelmä
3. Avainten muodostaminen
4. Asiakkaan todentaminen (valinnainen)

Oikea SSL: kättely (2)

1. Asiakas lähettää käyttämiensä algoritmien listan yhdessä haasteen (nonce) kanssa
2. Palvelin valitsee algoritmin listalta ja lähettää takaisin: valintatieto + varmenne + palvelimen haaste
3. Asiakas tarkistaa varmenteen, purkaa palvelimen julkisen avaimen esiin, generoi ennakkosalaisuuden (pre_master_secret, PMS), salaa sen palvelimen julkisella avaimella ja lähettää palvelimelle
4. Asiakas ja palvelin laskevat kumpikin erikseen salaus- ja MAC-avaimet ennakkosalaisuuden ja haasteiden avulla
5. Asiakas lähettää todennusosion (MAC), joka kattaa kaikki kättelyn viestit
6. Palvelin lähettää todennusosion (MAC), joka kattaa kaikki kättelyn viestit

Oikea SSL: kättely (3)

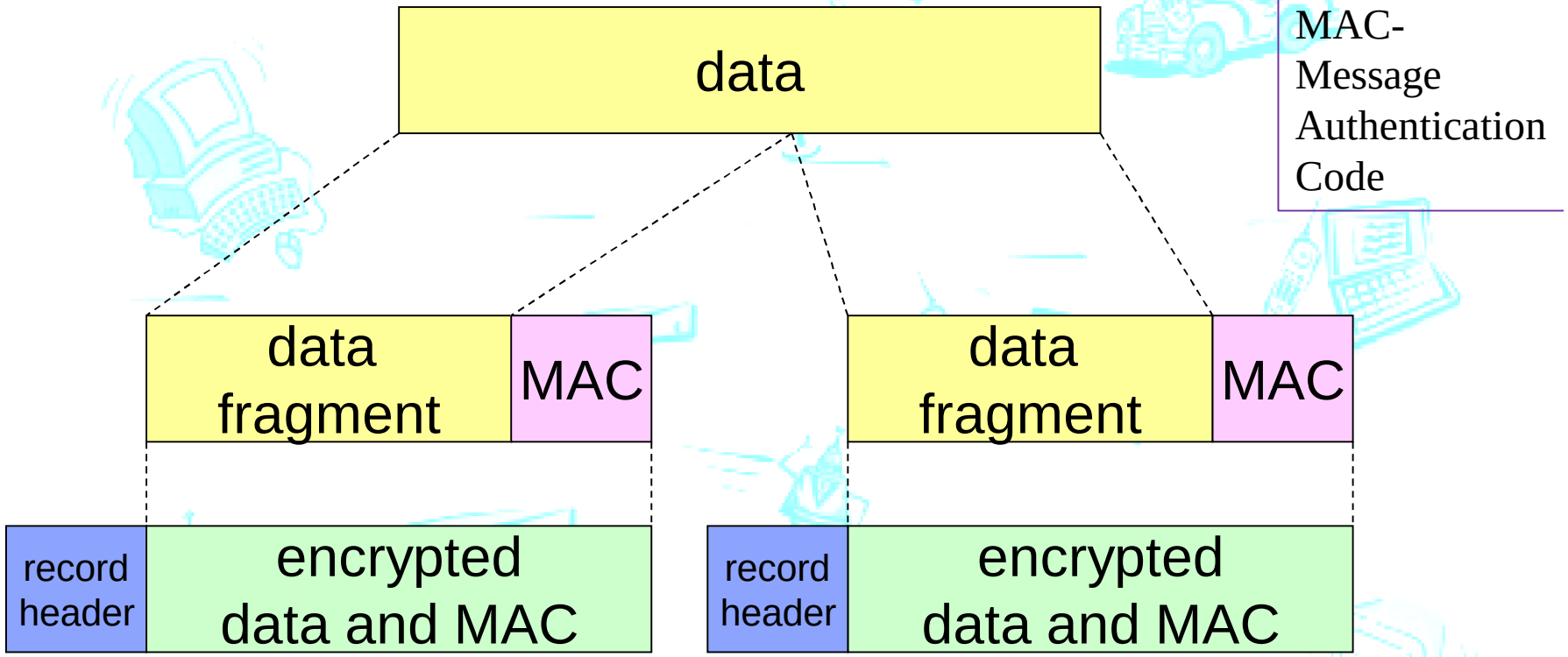
viimeiset 2 vaihetta suojaavat kättelyä hyökkäyksiltä

- Asiakas tarjoaa (selväkielisenä) algoritmijoukkoa, joista osa on vahvoja ja osa heikkoja suojausmenetelmiä
  - Heikko = helposti murrettava
- Matkalla man-in-the-middle voi poistaa vahvemmat algoritmit asiakkaan listalta
- Kaksi viimeistä vaihetta suojaavat lähetetyt viestit ja näin estävät niiden muuttamisen
  - Näiden kahden vaiheen viestit on jo salattu, aiemmat vaiheet selväkielisinä viesteinä.

Oikea SSL: kättely (4)

- Miksi kaksi haastetta (eli satunnaislukua)?
- Oletaan, että Trudy kuuntelee ja tallettaa kaikki viestit, joita Alice ja Bob lähettävät
- Myöhemmin Trudy ottaa TCP-yhteyden Bob:iin ja lähettää täsmälleen saman viestisekvenssin
  - Bob (esim. Amazon) olettaa, että Alice tekee toisen identtisen tilauksen samalle tuotteelle
  - Ratkaisu: Bob lähettää erilaisen satunnaisen haasteen jokaiselle yhteydelle. Näin eri yhteyksien salausavaimista tulee erilaiset
  - Bob voi nyt havaita todennusosion avulla, että vastapuoli ei ole oikea ja viestinnän yksityisyyttä on rikottu.

SSL tietueet (records)

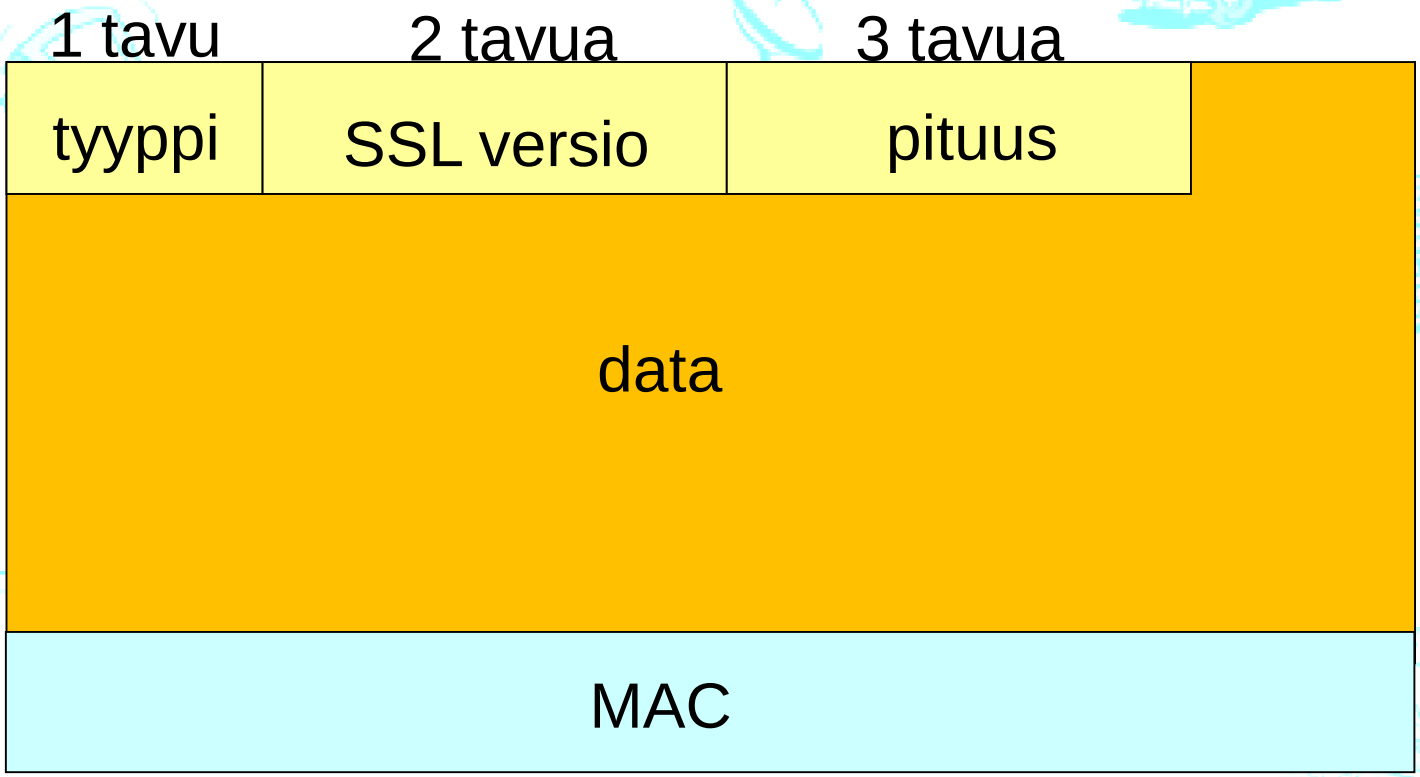


Tietueen otsake: tietueen tyyppi; versio; pituus

MAC: sisältää järjestysnumeron, MAC avain M_x

fragmentti: jokainen SSL osio 2^{14} tavua (~16 KB)

SSL tietueen rakenne



data ja MAC salattu (symmetrinen algoritmi)

Oikea SSL: yhteyden viestit

handshake: ClientHello

handshake: ServerHello

handshake: Certificate

handshake: ServerHelloDone

handshake: ClientKeyExchange

ChangeCipherSpec

handshake: Finished

Tästä eteenpäin kaikki salattu!

ChangeCipherSpec

handshake: Finished

application_data

application_data

Alert: warning, close_notify

TCP FIN seuraavaksi

VERKKOKERROS: IPSEC

Verkkokerroksen luotettavuus? Mitä/miksi?

Kahden verkkoelementin (esim. reitittimiä) välistä:

- Lähettävä elementti salaa hyötykuorman, joka voi olla:
  - TCP tai UDP segmentti, ICMP viesti, OSPF sanoma ….
- Kaikki data näiden kahden välillä jää piiloon:
  - Verkkosivut, sähköposti, P2P tiedostonsiirrot, TCP SYN, jne …
- Suojapeite “blanket coverage”

Virtuaalinen yksityisverkko (Virtual Private Network, VPN)

Motivaatio:

- instituutiot (yritykset, yliopistot, yms.) haluavat yksityisen verkon tiedon suojauksen takia.
  - Aito yksityinen verkko on kallis: erilliset reitittimet, linkit, fyysiset yhteydet, DNS arkkitehtuuri.
- VPN: Instituution toimipisteiden välinen yksityinen liikenne julkisen internetin kautta
  - Salataan ennen julkiseen internetiin laittamista
  - Loogisesti erotettu muusta liikenteestä

Virtuaalinen yksityisverkko (VPN)

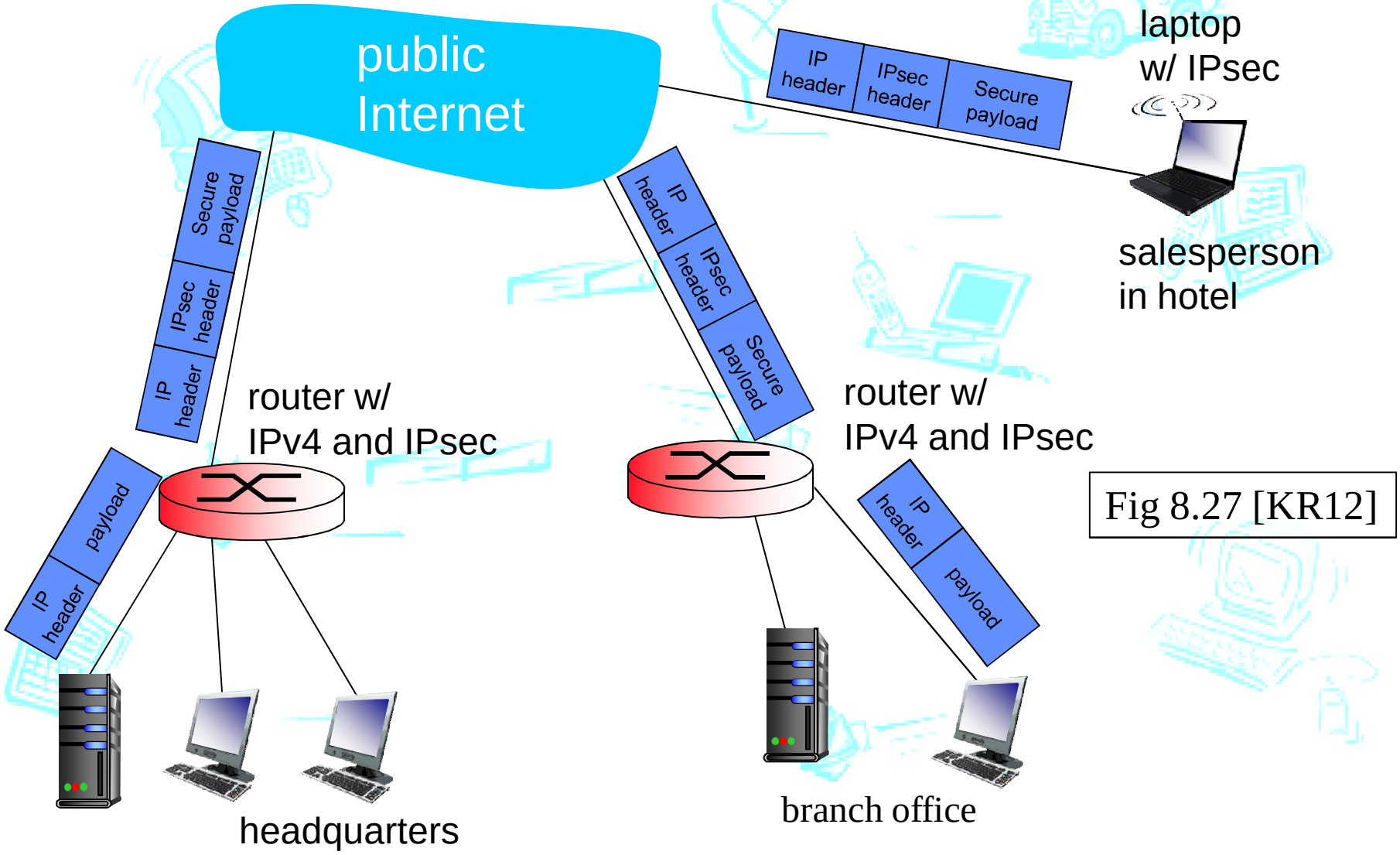
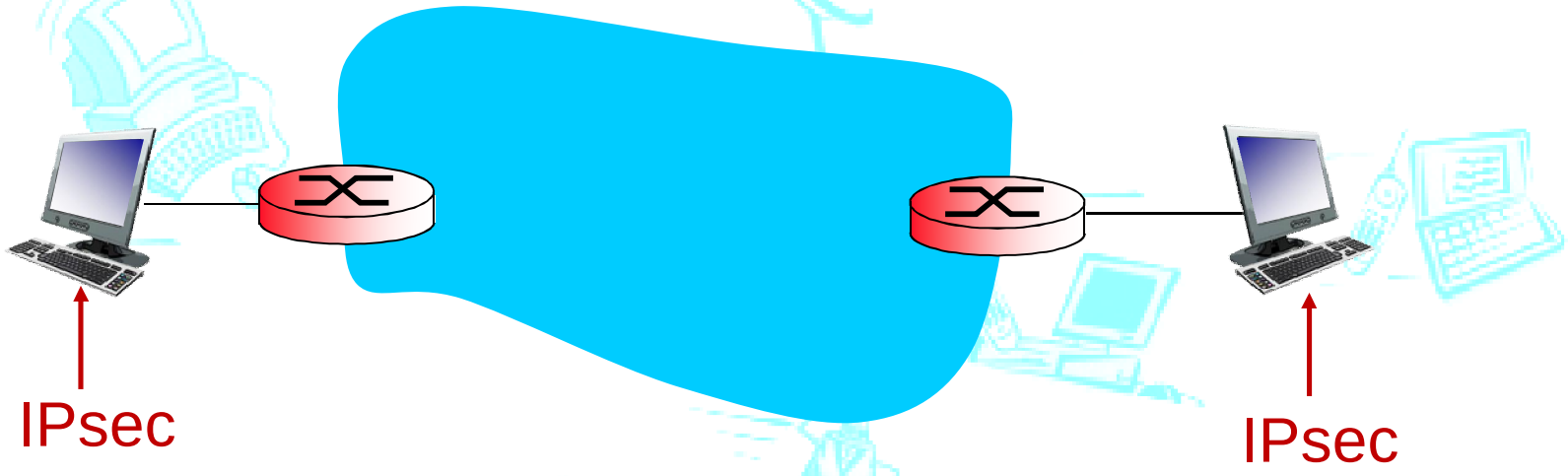


Fig 8.27 [KR12]

IPsec:n ominaisuuksia

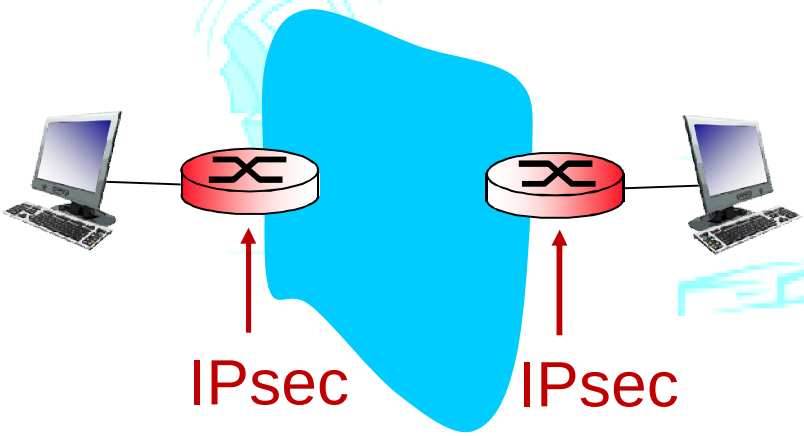
- Viestien eheys, yksityisyys
- Alkuperän (=lähettäjän) todentaminen
- Toistohyökkäysten estäminen
- Luottamuksellisuus
- Oikeastaan kaksi eri protokollaa, jotka tukevat näitä ominaisuuksia eri tavoin:
  - Authentication Header, AH - ei takaa luottamuksellisuutta (= data selväkielisenä)
  - Encapsulation Security Payload, **ESP** – takaa kaikki y.o. ominaisuudet, paljon yleisemmin käytetty kuin AH
- Kaksi eri toimintamuotoa: siirto (transport) ja tunnelointi

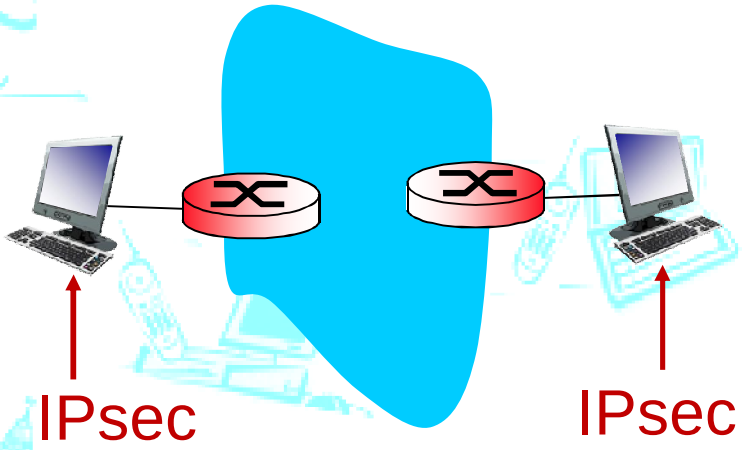
IPsec siirtomuoto (transport, host)



- Alkuperäiset isäntäkoneet (hosts) laativat IPsec datagrammit päästä-päähän yhteydelle
- Suojaa ylempien kerrosten protokollia
  - salaa vain datan, ei otsakkeita (ellei käytetä AH:ta)
- Ei käsitellä kirjassa eikä kurssilla

IPsec tunnelointi (tunneling)





- Organisaation internetiin yhdistävät reitittimet osaavat IPsec:iä

- ja/tai isäntäkoneet osaavat IPsec:iä

Neljä kombinaatiota!

Host mode with AH	Host mode with ESP
Tunnel mode with AH	**Tunnel mode with ESP**

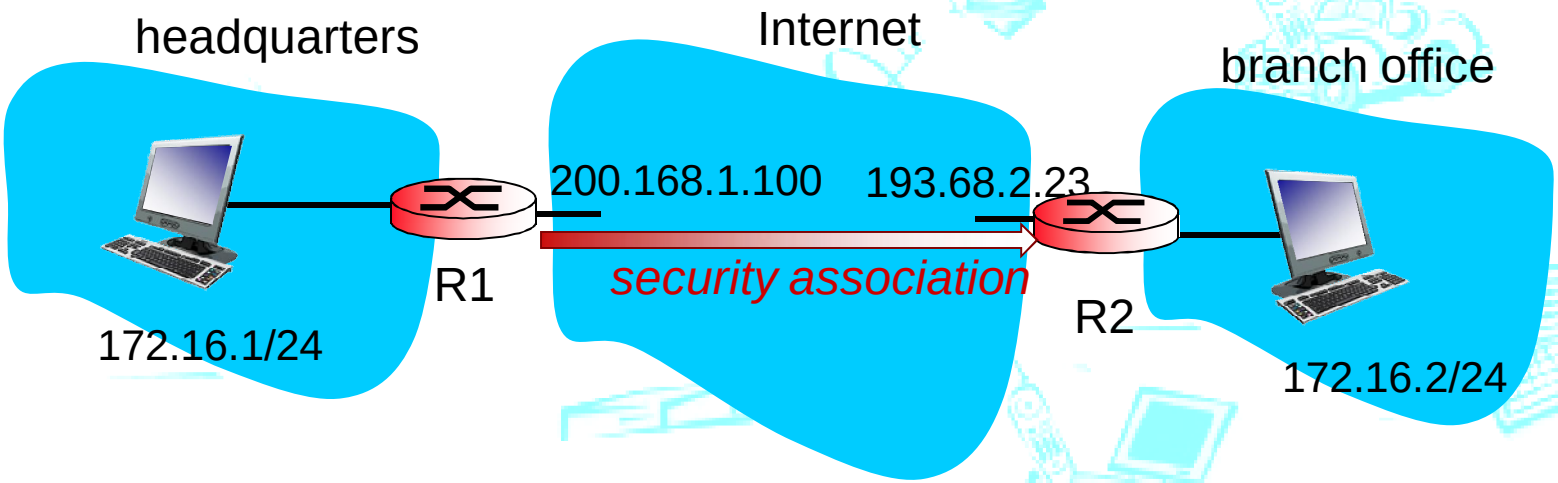
Yleisin ja tärkein!

AH - Authentication Header, AH
ESP - Encapsulation Security Payload

Turvayhteydet (Security associations, SAs)

- Ennen datan lähetystä muodostetaan "turvayhteys" (security association, SA) lähettäjältä vastaanottajalle
  - Turvayhteydet ovat yksisuuntaisia!
- Kommunikoinnin osapuolet ylläpitävät turvayhteyteen liittyvää *tilatietoa*.
  - Muistathan, että TCP:n osapuolet ylläpitävät tilatietoa
  - IP on tilaton; mutta siis IPSec:issä on tila!
- Yksisuuntaisuus: Kuinka monta turvayhteyttä tarvitaan virtuaalisessa yksityisverkossa pääkonttorin, sivukonttorin ja n:n myyntimiehen välillä?
  - 2 + 2n

Turvayhteys reitittimillä R1 ja R2



R1:n ja R2:n turvayhteyteen (SA) liittyviä tietoja:

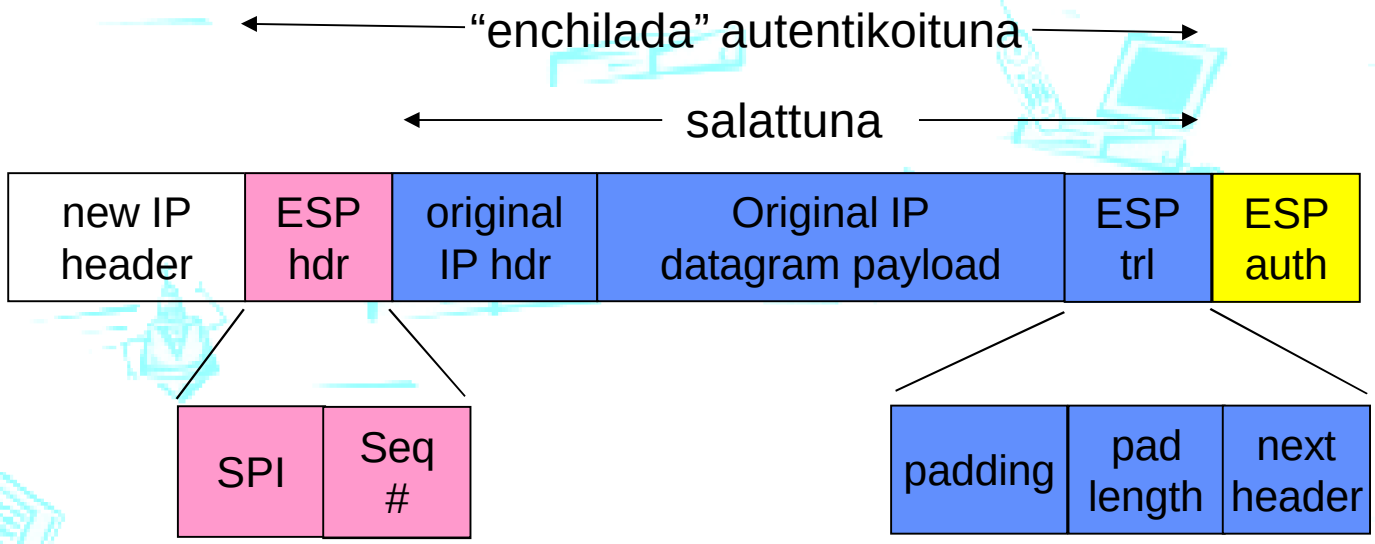
- 32-bittinen tunniste: *turvaparametri-indeksi (Security Parameter Index, SPI)*
- Lähettäjän (origin) SA rajapinta (200.168.1.100)
- Vastaanottajan (destination) SA rajapinta (193.68.2.23)
- Käytettävän salauksen tyyppi (esim. 3DES with CBC)
- Salausavain
- Käytettävän eheystarkistuksen tyyppi (e.g., HMAC with MD5)
- Tunnistusavain

Turvayhteyskanta (Security Association Database)

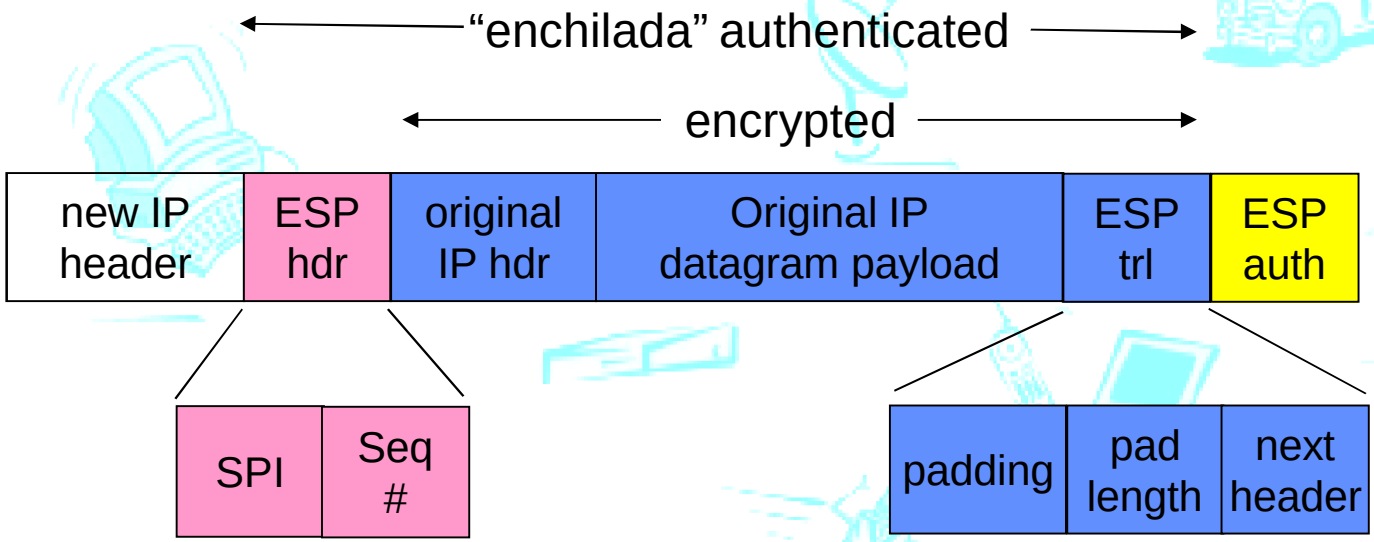
- Yhteyden päätepisteet tallettavat turvayhteyden tilan *turvayhteyskantaan (security association database, SAD)*, josta se on saatavilla toiminnan aikana.
- Kaikki turvayhteydet samassa kannassa
- Lähettäessään reititin R1 katsoo turvayhteyskannasta miten datagrammin kanssa pitää menetellä.
- Vastaanottaessaan IPsec datagrammin R2 katsoo turvayhteyskannasta turvaparametri-indeksin (SPI) osoittamasta paikasta, miten datagrammin kanssa pitää menetellä.

IPsec datagrammi

Tarkastellaan vain tunnelointitapaa ESP



Lähettäjä kokoaa paketin “enchilada”:



- ESP lopuke (trailer,trl): lohkosalauksen täytebitit
- ESP otsake (header,hdr):
  - Turvaparametri-indeksi SPI, jotta vastaanottaja tietää mitä tehdä
  - Järjestysnumero, joka estää toistohyökkäykset
- Todennusosio MAC (ESP auth –kentässä) muodostetaan käyttäen jaettua salaista avainta

IPsec järjestysnumerot (seq #)

- Uudelle turvayhteydelle lähettäjä alustaa seq. # = 0
- Aina kun turvayhteydellä lähetetään datagrammi:
  - Lähettäjä kasvattaa laskuria seq #
  - Laskurin arvo myös datagrammin kenttään seq #
- Tavoite:
  - Estetään ylimääräisen kuuntelijan toistohyökkäys
  - Kahdentunut, autentikoitu IP datagrammi voi sekoittaa palvelun
- Menetelmä:
  - Vastaanottaja tarkistaa kahdentumiset (ns. duplikaatit)
  - Vastaanottaja ei tarkkaile *kaikkia* saapuneita paketteja, vaan käyttää ikkunointia

Turvakäytännöt/-politiikat (Security Policy Database, SPD)

- Käytäntö (policy): Kullekin datagrammille lähettäjän täytyy tietää pitääkö sen käyttää Ipsec:iä
- Täytyy myös tietää mikä turvayhteys tähän liitttyy
  - Esim: lähettäjän ja vastaanottajan IP:t ja protokollanumero
- Tietokannasta SPD löytyy tieto siitä, mitä (what) saapuvalla datagrammille pitää tehdä
- Turvayhteyskannasta SAD löytyy tieto siitä, kuinka (how) tuo tehdään

IPsec - kuuntelija välissä

- Oletus: Trudy on jossain R1:n ja R2:n välillä, mutta hän ei tiedä avaimia.
  - Voiko Trudy nähdä alkuperäisen datagarammin sisällön?
  - Entä lähettäjän ja vastaanottajan IP:t, kuljetuskerroksen protokollan, sovelluksen käyttämän porttinumeron?
  - Voiko Trudy vaihtaa bittejä huomaamatta?
  - Tekeytyä R1:ksi käyttäen R1:n IP-osoitetta?
  - Toistaa datagrammin?

Avainten vaihto (Internet Key Exchange, IKE)

- *Edellä ei otettu kantaa jaettujen avainten määrittämiseen.* Sen voi tehdä manuaalisesti IPsecin päätepisteisiin:

  Example SA

  SPI: 12345
  Source IP: 200.168.1.100
  Dest IP: 193.68.2.23
  Protocol: ESP
  Encryption algorithm: 3DES-cbc
  HMAC algorithm: MD5
  Encryption key: 0x7aeaca…
  HMAC key:0xc0291f…

- Käytännössä manuaalinen avaintenhallinta ei toimi, jos VPN:ssä satoja päätepisteitä
- Siksi on olemassa *IPsec IKE (Internet Key Exchange)*

IKE: PSK and PKI

- Todentaminen (authentication) joko
  - Etukäteen jaetulla salaisuudella (pre-shared key PSK) tai
  - Julkisen avaimen salauksella PKI + serfikaateilla
- PSK: molemmat aloittavat jaetulla salaisuudella
- PKI: molemmat aloittavat sertifioiduilla julkisilla avaimilla (ja salaisella omalla avaimella)

- Varsinainen toiminta on kaksivaiheinen:
  - Vaihe 1: Käytetään IKE:ä toisen todentamiseen ja IKEn turvayhteyden luontiin (ISAKMP security association)
  - Vaihe 2: Neuvotellaan IPsecin avaimet (kaikki neljä) muodostetun IKEn turvayhteyden varassa.