

**9.a** Ismertesse a VPN-ek (Virtual Private Network) célját, feladatát és fajtáit! Milyen megvalósításait ismeri? Miben különböznek a különböző rétegekben megvalósított VPN-ek?

A VPN fogalma, rendeltetése, alaptípusai, funkciói, szolgáltatásai, topológiák

VPN – Virtual Private Network

- **Virtuális:** A magánhálózat forgalma nyilvános hálózaton halad keresztül egy virtuális alagúton.
- **Védett:** Átmenő forgalom titkossága biztosított.

Rendeltetése

- Biztonság növelése
- Anonimitás
- Nem elérhető tartalomhoz jutás (adott országon belül például tiltva van)
- Adatvédelem

Alaptípusai

- IPSec – Internet Protocol Security
- L2TP - Layer 2 Tunneling Protocol
- PPTP – Point-to-Point Tunneling Protocol
- SSL és TLS
- OpenVPN
- SSH – Secure Shell

Topológiák

- **Site-to-Site VPN**
  - o Két vagy több LAN kapcsolható össze.
  - o Az állomások normál IP csomagokat küldenek, ami egy VPN gateway-en megy keresztül.
- **Client-to-Site VPN:**
  - o Kliens-szerver kapcsolat, ahol kliens alkalmazás szükséges.
- **Client-to-Client VPN:**
  - o Közvetlen kommunikáció két számítógép között, központi szerver nélkül.

## A különböző OSI modell szerinti rétegekben széles körben elterjedt VPN megvalósítások jellemzői, előnyei és hátrányai

### L2 VPN

- Független a felső protokolltól
- Egy-egy kapcsolatot véd, így minden összeköttetésre külön alkalmazni kell.
- MITM támadás lehetséges

### L3 VPN

- Média és alkalmazás független
- IPSec, GRE, MPLS védelem

### L4 VPN

- SSL-lel biztosítja a titkosságot, a felhasználók hitelességét és az adatok sértetlenségét a TCP alkalmazások számára.
- Nem rugalmas, nehéz megvalósítani
- Nem alkalmazás független

### L7 VPN

- Az alkalmazás rétegbeli VPN-t minden alkalmazásban külön-külön meg kell valósítani.

## GRE kapcsolat szolgáltatásai, alkalmazási kör, jellemzők, „site-to-site” GRE konfigurálása

### GRE kapcsolat szolgáltatásai

- Hálózati protokollok közötti átjárás
- Többszintű hálózatok összekapcsolása
- Hálózatok közötti tűzfalak átjárásának lehetősége
- Hálózatok közötti VPN-ek létrehozása

### Jellemzők

- Nem alkalmaz titkosítást, így IPSec-et kell alkalmazni.
- Támogatja a routing protokollokat
- Több protokollal alagutakat is támogat
- Multicast csomagokat is kezel
- Alkalmaz irányító protokollok irányítási információinak szállítására és cseréjére.

# Az SSL protokoll célja és feladata, szerkezeti felépítése, alprotokolljai és feladataik

## SSL célja

- Titkosított kommunikációt biztosító protokoll, ami nyílt hálózatokban, kapcsolatorientált kommunikációban nyújt védelmet.
- Csak egy-egy kommunikációs csatornát biztosít.
- Gyakran használják a weboldalak biztonságos titkosítására is.

## SSL szerkezeti felépítése

- Minden egyes kapcsolat egyedi kulccsal titkosít.
- Tanúsítvány igazolja a szervert.
- Biztosítja az adatintegritást. (MD5, SHA-1)

## SSL működése

1. Kliens csatlakozik a kiszolgálóhoz.
2. Kiszolgáló elküldi a hitelesítési tanúsítványt a kliensnek.
3. Kliens ellenőrzi a tanúsítvány hitelességét, majd létrehozza a titkosított kapcsolatot a kiszolgálóval.
4. Kliens és kiszolgáló között így már biztonságosan lehet adatokat cserélni.
5. Ha az SSL kapcsolat megszakad, akkor a kliens és a kiszolgáló kapcsolata is megszakad.

## SSL alprotokolljai

### Rekord protokoll

- Feladata a kliens és a szerver és a felsőbb SSL protokoll entitások védelme:
  - Titkosítás, integritásvédelem, üzenet-visszajátszás elleni védelem

### Handshake protokoll

- Rekord protokollban használt kriptográfiai algoritmusok és paramétereik egyeztetése.
- Kulcscsere és hitelesítés

### Change-Cipher-Spec protokoll

- Egyetlen üzenetből áll, ami a Handshake protokoll kulcscsere részének végét jelzi.
- Ezt az üzenetet elküldi, utána az adott fél az új algoritmusokat és kulcsokat kezdi használni a küldése.
  - A vétel még mindig a Handshake előtti állapot szerint történik.

### Alert protokoll

- Figyelmeztető és hibaüzenetek továbbítása.

## A handshake, valamint a record alprotokoll feladata, működése és üzenetei

### Rekord protokoll működése

- A felsőbb protokoll rétegektől érkező üzeneteket:
  - o Fragmentálja, ha szükséges.
  - o Fragmenseket tömöríti
  - o Tömörített fragmenseket fejléccel látja el
  - o Fejléccel ellátott, tömörített fragmensre üzenethitelesítő kódot/MAC-et számol és azt a fragmenshez csatolja.
  - o Az üzenethitelesítő kóddal ellátott fragmenst rejtjelezi.

### Rekord üzenetei

- **type:** Rekord üzenetben melyik felsőbb protokoll található.
- **version:** SSL verzió
- **length:** Fragmens hosszát tartalmazza bájtban mérve.
- **MAC:** Üzenethitelesítő kód generálása

### Handshake protokoll működése

1. **fázis:** Kliens és szerver elküldi a tulajdonságait, megállapodnak
2. **fázis:**
  - a. Kulcscseremódszertől függ
  - b. Szerver elküldi a tanúsítványát és kéri a kliens tanúsítványát.
3. **fázis:** Tanúsítvány ellenőrzés és kulcscsere folytatása
4. **fázis:** Kulcscsere életbelépése, befejezése

### Handshake üzenetei

- **KliensHello:**
  - o Kliens küldi ezt az üzenetet az SSL Handshake kezdeményezésére.
  - o Kliens verzió, véletlenszám, viszonyazonosító, biztonsági algoritmusok, tömörítő algoritmusok
- **SzerverHello:**
  - o Kiszolgáló küldi a **KliensHello** üzenetre válaszul.
  - o Szerver verzió, véletlenszám, viszonyazonosító, biztonsági algoritmusok, tömörítő algoritmusok
- **Szerver kulcscsere üzenet**
- **Tanúsítvány kérés**
  - o Előfordulhat olyan eset is, amikor a tanúsító hatóságok listája üres.
    - Ilyenkor a kliens eldöntheti, hogy elküldi-e az ügyféltanúsítványt vagy sem.
- **Kliens tanúsítvány**
  - o A kliens bemutatja a tanúsítványláncát a kiszolgálónak.
- **Kliens kulcscsere üzenet**
  - o Lényege, hogy létrehozza a közös kulcsot a kliens és a kiszolgáló között anélkül, hogy azt egy kívülálló számára felfedné.
- **Kész üzenet**
  - o Első olyan üzenet, ami már az új algoritmusokat használva, az új kulcsokkal van kódolva.