IP biztonság – 2. rész

oktatási segédanyag az „IP alapú távközlés” c. tárgyhoz

Készült:

Csehi László András

IP minőség és biztonság

című szakdolgozatának felhasználásával.

Szerkesztette: Lencse Gábor

Az anyag témaköre:

Biztonsági megoldások a kommunikáció védelmére:

- VPN (MPLS, IPsec, OpenVPN)
- TLS
- SRTP

Tartalomjegyzék

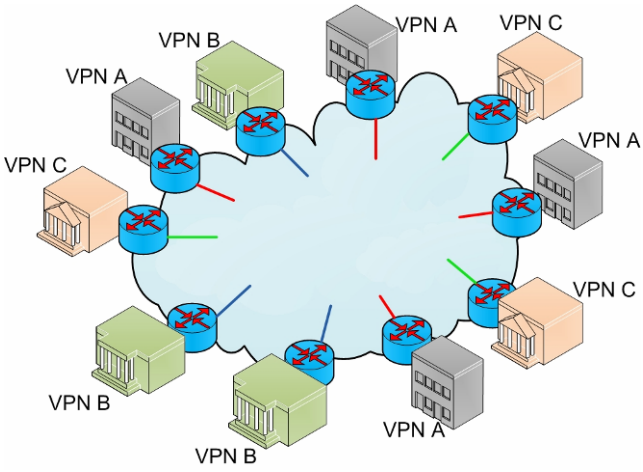
1. VPN-ek

1.1 MPLS VPN (MPLS Virtual Private Network)

Az MPLS VPN működése kissé eltér az ismertetett MPLS működésétől, ugyanis az egyes VPN-ek magán IP címtartományokat is használhatnak. Az IANA (*Internet Assigned Numbers Authority*) az alábbi három blokkot különítette el magánhálózatok számára:

- 10.0.0.0 – 10.255.255.255
- 172.16.0.0 – 172.31.255.255
- 192.168.0.0 – 192.168.255.255

A címtartományok két VPN hálózat között átfedhetik egymást. Az MPLS VPN eltérő *második rétegű* hálózatok (ATM, PPP, Ethernet, stb.) között is képes kialakítani trusted VPN-t. Ahhoz, hogy ezt leegyszerűsítve szemléltetni tudjuk, úgy kell értelmeznünk a VPN hálózatokat, mintha azok Ethernet VLAN-ok (*Virtual Local Area Network, virtuális helyi hálózat*) lennének [12]. Az **1. ábrán** látható VPN hálózatok közötti kapcsolatot egyrészt azért nem szemléltettük, hogy látni lehessen a VLAN-okkal való hasonlóságot, másrészt pedig bonyolult és különböző hálózati struktúrákat nem akartuk összevegyíteni egy kis ábrába. Az *Edge LSR*-ek a VPN tagok (A, B, C). A *Core LSR*-ek nem tudnak a VPN kialakításról, csupán az első *címke* alapján végzik el a kapcsolást. Magától értetődik, hogy egy *belső router* több VPN hálózatnak is a része lehet (közös erőforrásként szolgál), amit úgy lehet elérni, hogy egy *router*en belül több *virtuális router*t is létrehozunk, külön-külön *routing tábl*ával (*VRF*, *Virtual Routing and Forwarding, virtuális útválasztás és továbbítás*). Az *Edge LSR*-ekben az interfészeket egyesével rendelhetjük hozzá a VRF-hez (ugyanúgy, ahogy a VLAN-okat a *switch*-hez). A VPN azonosítására egy másik *címke* szolgál. Ez is belekerül a csomagba. Az MPLS VPN tagok között MBGP (*Multiprotocol Border Gateway Protocol, többprotokollos határ átjáró protokoll*) szállítja a VRF táblákat [12]. A VPN hálózatok még akkor sem látják egymást, ha



1. ábra *MPLS VPN*

közös erőforrást használnak. A telephelyek felől a hálózat úgy néz ki, mintha csak egy *útválasztó*n keresztül kommunikálnának egymással.

Az MPLS VPN a VPN szolgáltatás egyszerűsítése, előszeretettel használják cégek és szervezetek a távoli telephelyeik közötti kommunikációban az adatbiztonság megóvása érdekében. Különösen alkalmas különböző forgalmi típusok kezelésére (VoIP, VoD, web, FTP, stb.). Mivel az MPLS módszer viszonylag fiatal, alig tíz éves múltra tekinthet vissza, régóta üzemelő hálózatokban lehet, hogy néhány eszközt le kell cserélni ahhoz, hogy a VPN szolgáltatást igénybe lehessen venni, ugyanis nem minden *router* támogatja a *többprotokollos címkekapcsolás*t.

1.2 Secure VPN megvalósításához használható protokollok

A secure VPN kapcsolatok megvalósításait a következő szabványok és protokollok teszik lehetővé:

- PPTP (*Point-to-Point Tunneling Protocol, pont-pont alagútprotokoll*): a *Microsoft* által kifejlesztett protokoll, mely elsősorban nem több LAN hálózat, hanem ügyfelek és távoli hálózatok összekapcsolására való (pl.: a felhasználó az *Internet*en keresztül jelentkezik be vállalatának hálózatához[1]) [14]. *Tűzfalak* között nem szokták alkalmazni. Sok *Microsoft Windows* disztribúció tartalmazza, de sajnos a belső fejlesztés miatt voltak biztonsági hibák, amelyek csak akkor derültek ki, amikor a külső szakemberek kezelésbe vették. A javítások sem a legkielégítőbb eredménnyel szolgáltak, de tény, hogy a legelterjedtebb szabvány. De nem a legjobb.
- L2TP (*Layer 2 Tunneling Protocol, második rétegű alagútprotokoll*): a második legelterjedtebb technológia, a *Microsoft* és a *Cisco* közös fejlesztése, mely a PPTP protokoll továbbfejlesztése. Az ügyfél és a távoli hálózatok összekapcsolásán kívül képes több hálózatot is összekötni. *Második réteg*beli protokollként működik, de az *ötödik réteg*ben tevékenykedik. Nem nyújt *titkosítás*t és *hitelesítés*t, ezért IPSec csomagba ágyazva közlekedtetik. Használatát a *Windows 2000* és *Windows XP szerver*ek illetve *kliens*ek támogatják. A gyakorlati életben inkompatibilis a NAT (*Network Address Translation, hálózati címfordítás*) szolgáltatással, de ez nem a szoftvercégek hibája, hanem a szabványok összeférhetetlensége okozza.
- IPSec (*Internet Protocol Security, IP biztonsági protokoll*): elsősorban az *útválasztó*k

[1] A belső, tipikusan vállalati hálózatot *intranet*nek nevezik.

és *tűzfal*ak közé (és nem VPN-ekhez) tervezték, a forgalom biztosítása végett [14]. Az L2TP-vel együtt használva nagy biztonságú, *többprotokollos magánhálózat*okat is meg lehet vele valósítani. *Virtuális magánhálózat*okban csak akkor használható, ha csak IP forgalmat bonyolítanak le rajta. Előnye, hogy rugalmas, hátránya, hogy bonyolult.

- CIPE (*Crypto IP Encapsulation*): ez egy folyamatban lévő project szoftvere. Célja az IP *router*ek *titkosítás*ának felépítése. Támogatja a titkosított IP és UDP csomagok *alagutazás*át (*tunnelling*). További részleteket és a szoftver aktualizált verzióit a projekt honlapján találhatjuk meg: *http://sourceforge.net/projects/cipe-linux*

Az elméleti áttekintés után nézzük meg a VPN két legnépszerűbb megvalósítását, az *OpenVPN*-t és az *IPSec*-et.

1.3 OpenVPN

Az *OpenVPN* a *virtuális magánhálózat*nak egy nyílt forráskódú szoftveres megvalósítása. Ingyenes, szinte minden platformra fel lehet telepíteni (*Windows 2000/XP/Vista, Unix/Linux, OpenBSD, FreeBSD, NetBSD, Mac OS X, Solaris*). Az *Internet*en elég jó dokumentációkat találhatunk, amelyekből jól látszik, hogy könnyen paraméterezhető. Ugyan vannak kisebb biztonsági problémái, de ezek megfelelő kezelésével kielégítő eredményeket lehet elérni.

Fontos technikai tulajdonsága, hogy IP-IP *alagutazás*t használ. Telepítéskor létrehozza a *tunnel* (*TUN/TAP, alagút*) *eszköz*t, feltéve, ha a *kernel* (*rendszermag*) támogatja. A TUN/TAP *eszköz* (*device*) nem más, mint egy virtuális hálókártya, amely egy TCP vagy UDP porton keresztül képes akár egy egész alhálózatot továbbítani [16]. A *hitelesítés* és a *titkosítás* a szintén ingyenes, nyílt forráskódú *OpenSSL* protokoll segítségével történik. Az *OpenVPN szerver* az *OpenSSL*-lel létrehozott *kulcs*okkal tudja azonosítani a *kliens*t. Megoldható, hogy a *kliens*ek lássák egymást, de gyakorlatilag nem nagyon van rá szükség. Szükség esetén a *kliens*ek tevékenysége a bejelentkezéstől a kijelentkezésig, teljes körűen felügyelhető, sőt, egy naplófájlban kerül eltárolásra, azaz *log*olva van. Az *OpenVPN* nemcsak a *szerver*, hanem a *kliens* oldalon is futtatható szolgáltatásként. Ez történhet kézi indítással vagy automatikusan [17].

1.4 IPSec (Internet Protocol Security)

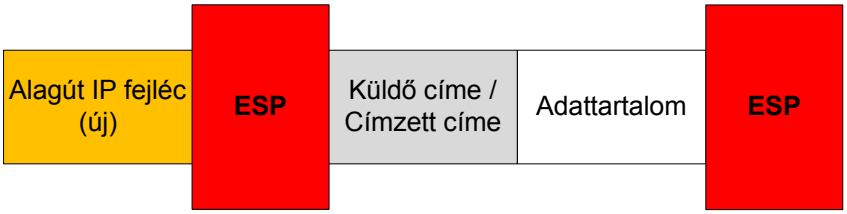
Az *OpenVPN* mellett az *IPSec* (*Internet Protocol Security, Internet biztonsági protokoll*) keretrendszer is nagy népszerűségnek örvend a VPN megvalósítások között. Olyan szabványhalmaz, amely a biztonságos IP *alagutazás*i (*tunnelling*) szolgáltatásokra vonatkozik. Az IPSec az *IP réteg*ére épül. *Virtuális magánhálózat*i szempontból három fontos kérdést taglal [13]:

1. *Hitelesítés*: arra szolgál, hogy a kommunikáló felek (személy, szervezet) biztosak lehessenek abban, hogy mind a két fél az, akinek mondja magát. *Virtuális magánhálózat*on belüli azonosításra, pontosabban *hitelesítés*re szolgál az IPSec szabvány *hitelesítési fejléc*e (*AH: Authentication Header*). Az *AH fejléc*et az IP csomaggal együtt szokták elküldeni. A *hitelesítés*en kívül biztosítja az *üzenetintegritás*t és az *ismétlésvédelmet* is. Az *üzenetintegritás*t úgy kell érteni, hogy az elküldött üzenet azonos a megkapott üzenettel, és az átvitel során nem módosították (külső támadás). Az ismétléssel elkövetett támadások jelentős károkat okozhatnak. A támadás egyszerűen csak egy már elküldött üzenet változtatás nélküli újraküldése a támadó részéről. Gondoljunk csak az ATM pénzjegykiadó automatáknál a pénzfelvételre vagy egy kereskedelmi árurendelésre. Az elsőnél nem mindegy, hogy ki veszi fel a pénzt, a másodiknál pedig az nem mindegy, hogy mekkora mennyiség után kell fizetni annak, akitől az árut várjuk. Ilyen és ehhez hasonló támadások kivédésére szolgál az *ismétlésvédelem*.
2. *Titkosítás*: vannak esetek, amikor az *Authentication Header* nem elég, mert *titkosítás*ra is szükség van. Az IPSec protokollcsalád *Encapsulating Security Payload* (*ESP, biztonságos beágyazott üzenettartalom*) összetevője gondoskodik a *titkosítás*ról. Ezt egy *szimmetrikus kriptográfiai algoritmus*sal (*DES, 3DES, Blowfish…*) valósítja meg annak érdekében, hogy az adatokat egy harmadik fél ne tekinthesse meg. Ugyanazok a tulajdonságai, mint az AH-nak. Az IPSec-et két üzemmódban szokták használni: az egyik a *szállítási mód* (*Transport Mode*, két számítógép közötti forgalom titkosítása), a másik az *alagút mód* (*Tunnel Mode*, két alhálózat között épít ki virtuális alagutat), ami már *virtuális magánhálózat*ot jelent. Az ESP-t a *Tunnel mód*ban alkalmazzák. Az eredeti *IP datagram* becsomagolásra kerül, és ez elé kerül egy *ESP fejrész* (ESP header), az elé pedig egy új *IP fejrész* (**2. ábra**). A teljes *IP datagram* végére is kerül egy *ESP farokrész* (ESP trailer). Jeff

Crume *Az internetes biztonság belülről* című könyvében azt a hasonlatot használja, hogy az IPSec *hitelesítési fejléc*e (AH) olyan, mint az ablakos boríték: látni lehet a címet, sőt, az AH esetén még az egész üzenetet is el lehet olvasni. Az *Encapsulating Security Payload* ezzel szemben egy közönséges zárt borítékhoz hasonlítható, amelybe belekerül az üzenet, így nem lehet látni se a tartalmát, se a végpontok címeit.

3. *Kulcskezelés*: a *küldő*nek (*sender*) és a *fogadó*nak (*receiver*) is rendelkeznie kell előre meghatározott *kulcs*okkal, amelyekkel *kódol*hatják és *dekódol*hatják az üzeneteket. A *kulcskezelés* tevékenységi körébe tartozik a *titkosító algoritmus* meghatározása, a *kulcscsere* gyakoriságának megadása, a titkosítás részleteinek biztonságos eljuttatása a kapcsolódni kívánó felekhez és egyéb folyamatok. Ebben segít az *Internet Key Exchange* (*IKE, Internetes kulcscsere*), ami a *biztonsági párbeszéd* (*Security Association, SA*) megvalósításának kereteit határozza meg, de nem definiálja azokat.

Az IPSec-nek is van egy ingyenes, nyílt forráskódú implementációja, a *FreeS/WAN*, de ez a projekt 2004-ben befejeződött. A *FreeS/WAN* projekt után mégis tovább folytatták a fejlesztéseket, és újabb lehetőség nyílt az IPSec (VPN) ingyenes megvalósítására



2. ábra *Az ESP alagút üzemmódja*

OpenS/WAN project néven. Mindkét megvalósítás alkalmas a *pont-pont* és az *ügyfél-kiszolgáló* (*client-server*) VPN összeköttetésekre. Igazából hálózatok összekapcsolására tervezték ezeket.

2. SSL/TLS (Secure Socket Layer / Transport Layer Security)

Az SSL (*Secure Socket Layer, biztonságos socket réteg*) és a TLS (*Transport Layer Security, szállítási rétegű biztonság*) biztonsági protokoll a nyilvános hálózatokon történő kommunikáció védelmére (*hitelesítés, titkosítás*) szolgál. TCP/IP hálózatokon futó alkalmazások két pontja (pl.: *szerver-kliens*) közti kommunikáció biztonságosságának megteremtésére használják.

Az IPSec-hez hasonlóan a *Secure Socket Layer* és a *Transport Layer Security* feladatai

közé tartozik a *hitelesítés* (*szerver-kliens*), az *adatintegritás* és *a titkosítás* [18]. Mivel mindez a *TCP réteg* fölött történik, ezért az IP rész adatai védtelenek maradnak, ami azt jelenti, hogy a végpontok kiléte nyilvánvalóvá válik.

2.1 TLS (Transport Layer Security)

A TLS két fontos protokollt használ:

1. *TLS Record Protocol*: amelynek a feladata a megbízható átvitelt biztosítása. *Szimmetrikus titkosítás*t (*DES, RC4…*) alkalmaz. Az adatokat *fragmentál*ja (a felső rétegből érkező adatfolyamot *tördel*i 2^{14} bájtnál kisebb darabokra), *tömörít*i, az *integritás*ukat megőrzi és *titkosít*ja [20].
2. *TLS Handshake Protocol*: amelynek a feladata, hogy a felek kapcsolatot tudjanak létesíteni egymással, azonosítani tudják egymást, és a megfelelő paramétereket (*session identifier, peer certificate, compression method, cipher spec, master secret, is resumable*)[2] egyeztetni tudják. Erre azért van szükség, hogy az átvitt rejtjelezett adatokat kezelni lehessen a kapcsolat másik végpontján.

A kapcsolat felépítésének folyamata az alábbi lépésekből áll [20]:

1. *Hello* üzenet cseréje, amelyben kicserélik a *véletlen kezdőszám*okat, a *titkosító algoritmus*t lefixálják és az *is resumable* kérdésére megadják a választ.
2. *Premaster secret* (*főkulcs előtti kulcs*) létrehozása a megfelelő *rejtjelezési algoritmus*ok kölcsönös cseréje után.
3. Az azonosításhoz szükséges *tanúsítvány*ok (*certificate*) és egyéb titkosítási információk elküldése egymásnak.
4. A *premaster secret kulcs*ból és a *véletlen szám*okból legenerálják a *master secret főkulcs*ot.
5. A *TLS Handshake Protocol* elküldi a titkosítási paramétereket a *TLS Record Protocol* részére.

[2] *Session identifier*: azonosító szám (szerver generálja a kliensnek). *Peer certificate*: végpont társ tanúsítványa. *Compression method*: tömörítési eljárás. *Cipher spec*: titkosító algoritmus. *Master secret*: 48 bites titkos kulcs (a szerver és a kliens ismeri). *Is resumable*: kapcsolat alapján engedélyezett-e új kapcsolat létrehozása.

3. SRTP (Secure Real-time Transport Protocol)

Az SRTP (*Secure Real-time Transport Protocol, biztonságos valósidejű átviteli protokoll*) az RTP egy olyan profilja, amely tud *titkosít*ani, *üzenetet hitelesít*eni és a *visszajátszás ellen* is *védelmet nyújt* az RTP és az RTCP forgalom számára [22]. Az RTCP biztonságos megfelelője az SRTCP (*Secure Real-time Transport Control Protocol*). Egy kis csoport alakította ki a *Cisco* és az *Ericsson* fejlesztői gárdájából.

Az *adatfolyam kódolás*ára és *dekódolás*ára az SRTP az SRTCP-vel együtt csak egyetlen egy rejtjelező szabványt használ, mégpedig az AES-t (*Advanced Encryption Standard, haladószintű titkosító szabvány*). Az AES két *rejtjelezési mód*ban működik:

- *Segmented Integer Counter Mode* (*osztott egész szám számláló mód*): egy tipikus *számláló mód*, ami véletlen hozzáférést biztosít bármelyik blokkhoz, amelyik fontos egy megbízhatatlan, csomagvesztésekkel járó hálózat RTP forgalomáramlásának szempontjából.
- *f8-mode* (*f8 mód*): a *kimenet visszacsatolási mód* (*Output Feedback Mode*) egy variációja. A *titkos-* és a *sózott kulcs* (*salted key*) alapértelmezett értékei ugyanazok, mint az AES *számláló mód*jában (*Counter Mode*).

Az SRTP biztosítani tudja az *adat*ok *integritás*át, a *hitelesítés*t és a *visszajátszás elleni védelmet* is. Az *üzenet hitelesítés*ére és tartalmának védelmére a HMAC-SHA1[3] algoritmust használja. Az SRTP egy *külső kulcs menedzsment protokoll*ra bízza a *kezdeti mesterkulcs* beállítását. Létezik két olyan protokoll, amelyeket speciálisan arra terveztek, hogy az SRTP-vel használják. Az egyik a ZRTP[4] (*Phil Zimmermann's Real-time Transport Protocol, Phil Zimmermann valósidejű átviteli protokollja*), a másik a MIKEY (*Multimedia Internet Keying, multimédia Internet „kulcshasználat"*).

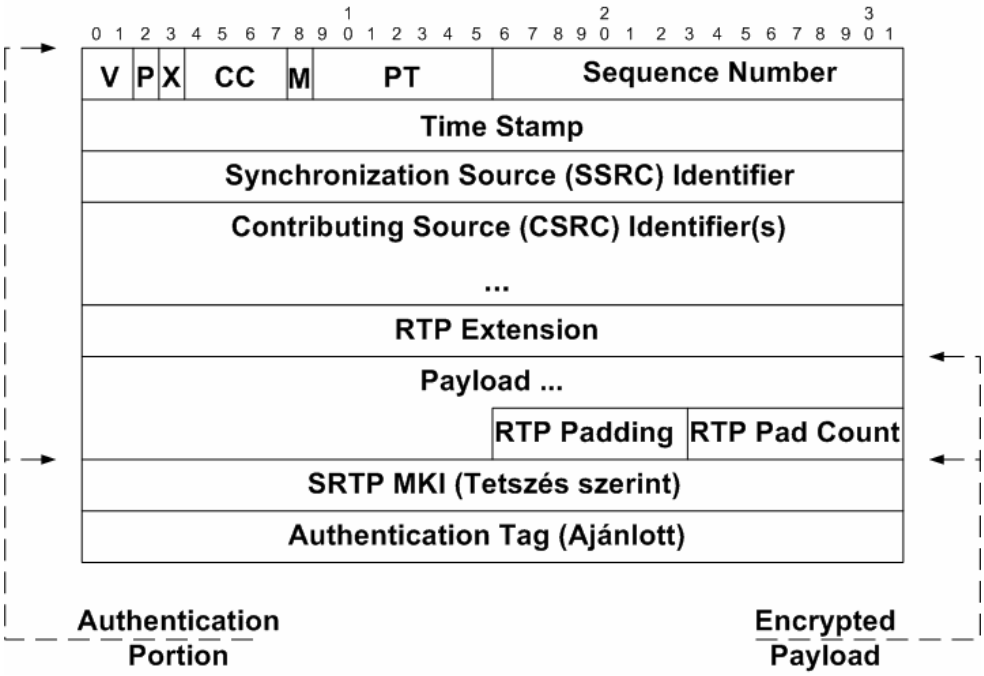
Ha az *SRTP fejrész*ét megnézzük (**3. ábra**), láthatjuk, hogy az *RTP fej*rész egésze megtalálható benne, és tartalmaz néhány új mezőt is. A magyarázatra szoruló mezők:

- *RTP Extension* (*Real-time Transport Protocol Extension): valósidejű átviteli protokoll kibővítése* (tetszés szerint)
- *Payload*: *hasznos teher*, például audió és videó (*n* x 64 bit, ahol *n* egész szám)

[3] HMAC: Keyed Hashing for Message Authentication (*kulcsos titkosítás üzenethitelesítéshez*): a MAC (Message Authentication Code Algorithm) üzenethitelesítő-kód algoritmus egyik megvalósítása.
SHA-1: Secure Hash Algorithm (*biztonságos titkosító algoritmus*): 160 bites egyirányú függvény.
[4] ZRTP: *Phil Zimmermann* (IETF) fejlesztette ki 2006-ban *Jon Callas* és *Alan Johnston* közreműködésével. A *Diffie-Hellman kulcscsere algoritmus* egy megvalósítását írja le.

- RTP Padding: RTP kitöltés (változó hosszúságú)
- RTP Pad Count: RTP kitöltés számláló (változó hosszúságú)

- *SRTP MKI* (*SRTP Master Key Identifier*): *SRTP mesterkulcs azonosító*, a kulcsmenedzsment határozza meg, jelöli ki és használja. Állítható hosszúságú (opcionális).
- *Authentication Tag*: *hitelesítési sor*, szintén állítható a hosszúsága. Erősen ajánlott a használata, ugyanis az üzenethitelesítés adatait hordozza.



3. ábra *SRTP PDU felépítése*

Az *Authentication Portion* magyarra fordítva azt jelenti, hogy *hitelesítési rész*. Az *RTP fejrész* és a *Payload* mezők együttesen alkotják a *hitelesítési rész*t. A *titkosított rész* (*Encrypted Payload*) maga a *hasznos teher*, ami valamilyen multimédia adat lehet (hang és/vagy videó).

4. Irodalomjegyzék

[12] Győri Gábor: *MPLS-VPN-ek a gyakorlatban*, LIAS-NETWORX Hálózatintegrációs Kft.
https*://nws.niif.hu/ncd2001/docs/eloadas/98/index.htm*

[13] Jeff Crume: *Az internetes biztonság belülről ...amit a hekkerek titkolnak*, Szak Kiadó 2003.

[14] Peter Norton, Mike Stockman: *A hálózati biztonság alapjairól*, Kiskapu Kft. 2000

[15] Dr. Lencse Gábor: *Hálózatok biztonsága*, 2008. Győr
http*://www.tilb.sze.hu*

[16] HupWiki - *OpenVPN*
http*://wiki.hup.hu/index.php/OpenVPN*

[17] HupWiki – *Az OpenVPN finomhangolása*
http*://wiki.hup.hu/index.php/Az_OpenVPN_finomhangol%C3%A1sa*

[18] Informatikai és Hírközlési Minisztérium, International Business Machines Corporation Magyarországi Kft. : *SSL/TLS alapú biztonságos kommunikáció*, 2004.
http://*www.itktb.hu/Resource.aspx?ResourceID=docstorefile&f=725&t=stored*

[19] Wikipédia – *Https*
http*://hu.wikipedia.org/wiki/Https*

[20] Bilicki Vimos: *Jegyzet a Fejlett Webes technológiák előadáshoz*, 2002.
http*://www.inf.u-szeged.hu/~bilickiv/fpt_2006_1/jegyzet/WebTechnologiak.pdf*

[21] Turányi Zoltán Richárd: *Hálózati trendek*, 1996. Budapest
http*://www.szabilinux.hu/trendek/trendek75.html*

[22] Wikipedia – *Secure Real-time Transport Protocol*
http*://en.wikipedia.org/wiki/Secure_Real-time_Transport_Protocol*