Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Bakalářská práce

2009 Zdeněk Kittler

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Podpora bezpečného tunelování v sítích IPv6 v operačních systémech

Zdeněk Kittler

Bakalářská práce
2009

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Katedra informačních technologií
Akademický rok: **2008/2009**

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Zdeněk KITTLER**

Studijní program: **B2646 Informační technologie**

Studijní obor: **Informační technologie**

Název tématu: **Podpora bezpečného tunelování v sítích IPv6 v operačních systémech**

Zásady pro vypracování:

Práce bude popisovat aktuální stav podpory bezpečnostního protokolu IPsec v IPv6 sítích v jednotlivých operačních systémech. - teoretické části práce popíše autor protokol IPsec a rozdíly mezi jeho použitím v IPv4 a IPv6 sítích - praktická část se bude skládat z otestování funkčnosti protokolu ve vybraných operačních systémech a následná analýza stavu podpory (podporované funkce, interoperabilita...) - nedílnou součástí práce bude zpracování návodů na zprovoznění tunelu v IPv6 sítích pod vybranými operačními systémy.

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

Satrapa, Pavel: Internetový protokol IPv6, CZ.NIC, Praha, 2008, ISBN 978-80-904248-0-7 Kent, S.: Seo, K.: Security Architecture for the Internet Protocol (RFC 4301), IETF, 2005

Vedoucí bakalářské práce: **Ing. Tomáš Fidler**
Katedra softwarových technologií

Datum zadání bakalářské práce: **15. ledna 2009**

Termín odevzdání bakalářské práce: **15. května 2009**

Univerzita Pardubice Česká republika

L.S.

doc. Ing. Simeon Karamazov, Dr.
děkan

Ing. Lukáš Čegan
vedoucí katedry

V Pardubicích dne 31. března 2009

Prohlašuji:

Tuto práci jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 15.05.2009

Zdeněk Kittler

Poděkování

Rád bych poděkoval vedoucímu bakalářské práce, Ing. Tomáši Fidlerovi, za rady, připomínky a návrhy týkající se bakalářské práce a za odborné vedení při provádění praktické části.

Anotace

Tato práce je zaměřena na analýzu současného stavu zabezpečení v IPv6 sítích pomocí VPN tunelů. Dále na srovnání stávajících možností zabezpečení na IPv4 a jejich případné porovnání se zabezpečením na sítích nové generace typu IPv6. Praktická část se zabývá implementací zabezpečení pomocí IPsec v různých operačních systémech a otestování funkčnosti v různých režimech přípustných danou implementací protokolu v systému.

Klíčová slova

IPV4; IPv6; IPsec; VPN tunelování; SSH; SSL; zabezpečený přenos

Title

Support for secure tunneling of IPv6 networks in operating systems

Annotation

This work is focused at analyzing the current state of security in IPv6 networks via VPN tunnels. Furthermore, compared to the existing security options for IPv4 and any comparison to the security of the network a new generation type of IPv6. The practical part deals with the implementation of security using IPsec in a variety of operating systems and test functionality in different modes permitted the implementation of the Protocol of the system.

Keywords

IPv4, IPv6, IPsec, VPN tunneling, SSH, SSL, secure transmission

Obsah

Seznam obrázků

Seznam tabulek

Seznam zkratek

AH (Authentication Header) – ověřovací hlavička

ARP (Address Resolution Protocol) – adresní protokol

ATM (Asynchronous Transfer Mode) asynchronní přenosový mód

CIDR (Classless Inter-Domain Routing) – třídní routovací protokol

DHCP (Dynamic Host Configuration Protocol) – dynamický konfigurační protokol pro hosty

DNS (Domain Name systém) – doménový jmenný systém

DOS (denial-of-service)- popírání služeb

ESP (Encapsulating Security Payload) – bezpečné zapouzdření dat

FTP (File Transfer Protokol) – Souborový transportní protokol

GRE (Generic Routing Encapsulation) – generické routovací zapouzdření

HTTP (HyperText Transfer Protocol) – hypertextový přenosový protokol

IANA (Internet Assigned Numbers Authority) – internetová autorita pro přiřazení čísel

ICCANN (Internet Corporation for Assigned Names and Numbers) – internetová korporace pro přidělování jmen a čísel

ICMP (Internet Control Message Protocol) – internetový kontrolní protokol pro zprávy

ICV (integrity check value) – hodnota kontrolující integritu

IETF (Internet Engineering Task Force) – internetová inženýrský skupina

IGMP (Internet Group Management Protocol) – internetový skupinový protokol pro údržbu

IKE (Internet Key Exchange) – internetový výměník klíčů

IP (Internet Protokol) - internetový protokol

IPv4 (Internet Protokol v 4) - internetový protokol verze 4

IPv6 (Internet Protokol v 6) - internetový protokol verze 6

IPsec (IP security) – IP bezpečnost

ISAKMP (Internet Security Association and Key Management Protocol) – protokol internetové bezpečnostní asociace a správce klíčů

L2TP (Layer 2 Tunneling Protocol) – tunelovací protokol na druhé vrstvě

LANE (LAN Emulation) – emulace Lan

LIR (Local Internet Registry) – lokální internetový registrátor

MPLS (Multiprotocol Label Swapping) – více protokolů přepínaných

MPOA (Multiprotocol over ATM) – více protokolů přes ATM

NAT (Network Address Translation) – Síťový adresní překladač

PPP (Point-to-Point Protocol) – bod bod protokol

PPTP (Point to Point Tunneling Protocol) – tunel protokol z bodu do bodu

RARP (Reverse Address Resolution Protocol) – reverzní adresovací protokol

RFC (Redy for comment) – žádost o komentáře

RIR (Regional Internet Registry) -regionální internetový registrátor

SA (Security Association) – Bezpečnostní asociace

SPD (Security Policy Database) – Bezpečnostní databáze politiky

SPI (Security Parameters Index) – Bezpečnostní index

SSH (Secure Shell) – bezpečný shell

SSL (Secure Socket Layer) – bezpečný socket

TCP (Transmission Control Protocol) – řídící přenosový protokol

TLS (Transport Layer Security) – transportní zabezpečená vrstva

TTL (Time to Live) – čas života

UDP (User Datagram Protocol) – uživatelský datový protokol

VPN Virtual Private Network) – virtuální soukromá síť

1 Úvod

Problematika zabezpečení přenášených dat, přihlášení do vzdáleného systému, sítě, nebo jen přenos souborů pomocí některého z transportních protokolů, je v dnešní době často diskutovaná věc, na kterou mnohé společnosti vykládají nemalé finanční prostředky. A proto se dnes nalézá mnoho, ať komerčních, nebo open source programů, které tuto problematiku řeší více či méně úspěšně. Během této bakalářské práce bylo testováno zabezpečení protokolem IPsec v různých operačních systémech. Všechny testy byly uskutečňovány jak v síti typu Ipv4, tak v IPv6 (pokud to daná implementace umožňovala). V dalším bodě jsou zkušenosti s použitím IPsec a to jak s konfigurací, tak i s provozem, porovnány s ostatními možnostmi zabezpečení. Během implementace protokolu IPv6 se často vyskytl problém s nepříliš dobrou podporou ze strany aplikací a značnou nepřipraveností poskytovatelů internetu. Další problémy činila ještě doposud neúplná, případně nepříliš vydařená implementace protokolu Ipv6, tak i samotný protokol IPsec, který se stále ještě nenachází ve finální verzi. I přes tyto nedostatky má tento druh zabezpečení jistou budoucnost a to ať z hlediska již stávající podpory ze strany komerční sféry (Cisco), tak i open source, a to především na operačních systémech typu Unix a Linux, kde je implementace již v dnešní době vydařená a dokonce je součástí samotného jádra systému Linux. Dalším místem pro použití VPN tunelů je cloud computing, který zaznamenává v současnosti bouřlivý rozvoj. A jistě bude i do budoucna velkou hybnou silou pro rozvoj jak síťových technologií, tak i softwarového vybavení osobních počítačů. V příloze bakalářské práce jsou návody na zprovoznění protokolu IPsec ve vybraných operačních systémech a to jak v IPv6 tak IPv4.

2 IP sítě

2.1 Definice IP sítí

Pojem IP sítě vychází z protokolu TCP/IP (Transmission Control Protocol / Internet Protocol), což je soubor protokolů využívaných pro komunikaci v počítačových sítích. Je to nejpoužívanější systém komunikace mezi počítači, jenž ve většině případů využívá i Internet. Ale tomu nebylo tak vždy. Jeho prvotní využití bylo určeno pro spojení několika vládních počítačů v síti Arpanet (Advanced Research Projects Agency. V 80tých létech, a to i díky značné kompatibilitě s mnoho hardwarovými a softwarovými systémy, se začal prosazovat v systémech typu Unix. Později ho za své přejali i jiní vývojáři, například pro systém od Microsoftu MS-DOS a s ním spojený OS Windows nebo OS/2. Tento protokol je i nadále vyvíjen a je do něho implementována řada nových služeb, které předchozí verze nepodporovaly.

2.2 TCP (Transmission Control Protocol)

TCP je to protokolem Transportní vrstvy, kde jeho hlavní činností je převádět odesílaná data do sekvence paketů a zároveň na protější straně doručené pakety ve správném pořadí složit a vytvořit z nich původní data. Tedy jedná se o spolehlivý přenos. Ten je dosažen očíslováním každého paketu a při nedoručení paketu nebo jeho poškození při přenosu zpětnou žádostí o znovuodeslání (kontrola poškození se provádí kontrolním součtem). Dále zajišťuje přeskládáváním paketů jejich doručení ve správném pořadí. K rozlišení komunikujících aplikací slouží v TCP takzvané porty, kde každá z komunikujících stran má přiřazeno číslo portu, jež může nabývat hodnot od 0 do 65535. Některé z těchto portů jsou již přiřazeny pro nejběžnější služby např. FTP(port 21), HTTP(port 80) nebo DNS(port 53) a díky tomu se dané služby nedostanou do konfliktu s jinou aplikací, která by daný port chtěla

použít pro komunikaci. Z tohoto důvodu jsou porty rozdělené do následujících kategorií:

– známé porty v rozsahu 0 až 1023

– registrované porty 1024 až 49151, kde by se měla čísla portů registrovat u ICANN (Internet Corporation for Assigned Names and Numbers)

– dynamické porty a porty pro soukromé použití 49152 až 65535.

2.3 IP (Internet Protokol)

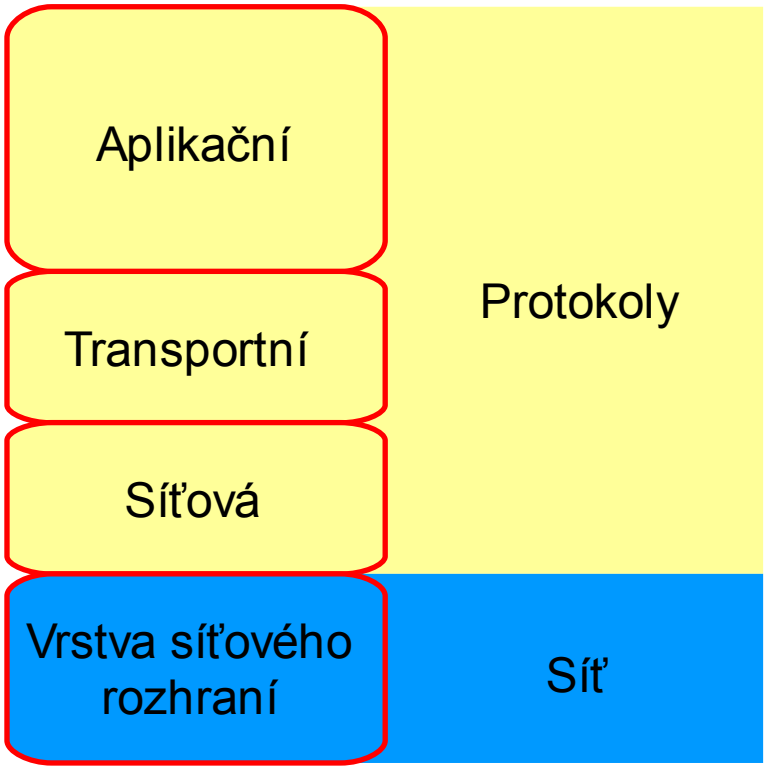
Jedná se o protokol síťové vrstvy, který se používá pro přenos dat skrz paketové sítě. Z hlediska přenosu se jedná o nespolehlivé doručení. To znamená, že strana, která odesílá data nepožaduje ověření o jejich doručení. Dokonce se ani nekontroluje doručení ve správném pořadí. Zde se nabízí využití některé z vyšších služeb např. TCP. Hlavní funkcí IP není starat se o data jako taková, ale o jejich doručení konkrétnímu uživateli (PC) v síti. To je dosaženo tím, že každému počítači v síti je přidělen unikátní identifikátor a tím je IP adresa. Podle něho dokáže každý počítač provést rozhodnutí, jak s daty naložit, zda je zahodit, odeslat dál či přijmout. Tyto možnosti síťových prvků se označují jako Routing (směrování). V běžném prostředí sítě se nejčastěji využívají zařízení nazývané Routery. V dnešní době je nejrozšířenější verzí IP protokolu verze s označením IPv4 , která bohužel i přes řadu opatření bude muset být nahrazena novější verzí a to IPv6 z důvodu nedostatku unikátních IP adres, které je možné přidělit počítačům v Internetu.

2.4 Architektura TCP/IP

Z důvodu požadované širší použitelnosti TCP/IP protokolu byl tento rozdělen do čtyř vrstev. Tyto vrstvy mají mezi sebou přesně definovanou formu

komunikace, tedy jedna vrstva může využívat službeb vrstev ostatních. Přesněji řečeno každá vrstva využívá služeb vrstvy nižší a nabízí své služby vrstvě vyšší. To umožňuje, aby spolu komunikovaly zařízení na stejných vrstvách skrz spojení vytvořené a obsluhované nižší vrstvou, bez ohledu na to, jaký protokol byl pro tuto činnost použit. Díky tomu může být TCP/IP nasazeno na mnoho fyzických přenosových technologií např. sériová linka, ethernet …

Na obrázku obrázku 1 je zobrazen čtyřvrstvý model TCP/IP.



Obrázek 1: Model TCP/IP

První vrstva „směrem“ od uživatele je aplikační vrstva (application layer). Tato vrstva má na starosti obsluhu požadavků procesů nebo celých programů pro přístup k síti a posílání či přijímání jejich dat od vrstev nižších. Konkrétně se může jednat například o tyto služby FTP, HTTP, DHCP atd. Na rozdíl od ISO/OSI modelu již neobsahuje podporu pro šifrování či kompri-

maci apod. Tato vrstva počítá s tím, že zmíněné nadstandardní služby mají být implementovány v aplikaci.

Pro rozlišení aplikačních protokolů se používají tzv. Porty. Ty umožňují rozlišit komunikující aplikace díky tomu, že každé spojení má určeno číslo portu, transportní protokol a IP adresu počítače. Dále mají Aplikační protokoly na výběr ze dvou režimů transportní vrstvy pro přenos dat a to buď pomocí spojitého TCP nebo nespojitého UDP. Některé aplikace využívají obou protokolů zároveň a to např. DNS.

Další vrstvou je transportní vrstva (transport layer). Je vrstvou jenž se implementuje na koncových zařízeních, kterými jsou počítače a přidává možnost svůj přenos zabezpečit a to z hlediska doručení odeslaných dat. Toho je možné dosáhnout použitím protokolu TCP, který je spojitý a požaduje ověření zda byla data doručena. Zmíněná vlastnost ale není pro všechny aplikace a služby jím poskytované důležitá, proto nabízí i možnost odesílat pakety bez ověření a to pomocí protokolu UDP. Příklad služby, která využívá UDP, může být streamované video, kde nám jde především o plynulý běh snímku ve stanoveném pořadí, než o 100 % stav každého ze snímků.

Třetí vrstvou je síťová vrstva (network layer). Tato vrstva by měla umožňovat, již z konceptu, co nejvyšší přenosovou rychlost a proto se nestará (data odesílá nespojitě) o to, zda byla data doručena. Tuto problematiku přenechává vrstvám vyšším. Hlavním úkolem síťové vrstvy je adresace v síti, směrování a předávání datagramů na další zařízení v síťovém prostředí. K tomu jí pomáhají další protokoly ICMP (Internet Control Message Protokol), ARP (Address Resolution Protokol), RARP (Reverse Address Resolution Protokol) atd.

Poslední z vrstev obsažených v TCP/IP modelu je vrstva síťového rozhraní (network interface). Právě tato vrstva tvoří zmiňovaný protokol tak

univerzální a to díky tomu, že nikde není specifikováno, jaké přenosové médium bude využívat, jakou rychlostí tato cesta bude disponovat, zda se jedná o cestu kvalitní či se špatnými přenosovými podmínkami. Z předchozího popisu plyne, že právě tato vrstva umožňuje přístup k fyzickému médiu a činí TCP/IP maximálně modulární.

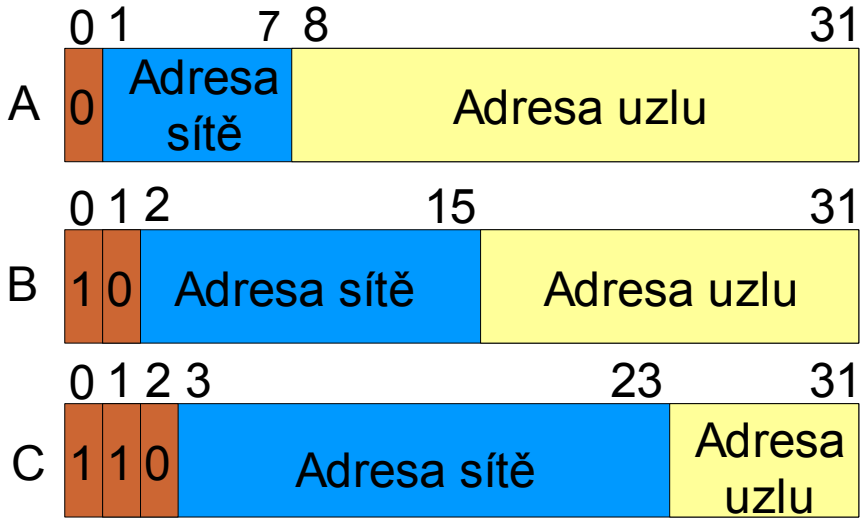
2.5 Bezpečnost v IP sítích

Hlavní předpoklad při vývoji TCP/IP byl vytvoření sítě, která by v době války umožňovala komunikaci mezi vzdálenými počítači a to i v případě vyřazení několika z nich. Proto pojem jako bezpečnost nebyla zahrnuta v důležitých parametrech, které by dané řešení mělo obsahovat. A proto není zabezpečení pomocí šifrování nebo jakéhokoliv jiného způsobu, implementováno. To v dnešní době způsobuje značné problémy a musí se řešit pomocí jiných aplikací, které si data zašifrují samy.

2.6 Sítě IPv4

Původ protokolu, jak již bylo zmíněno, je v síti ARPANET, která měla za úkol propojit počítače ministerstva obrany pro vědecké a výzkumné účely. Zde se ještě před protokolem IP používal protokol NCP (Network Control Program). Ten byl z důvodů značných nedostatků v roce 1983 nahrazen nově vyvinutým protokolem IPv4. Finální verzi IPv4 ještě předcházely verze nižší a to IPv0 až IPv3. Ty se nikdy nepoužívaly a proto jediné místo, kde se s nimi můžeme setkat, jsou laboratorní podmínky. Další verze, která není příliš známá, je verze s číslem 5. Ta, již s původním účelem, byla vyvinuta pro experimentální pokusy, konkrétně jako stream protokol. Měl koexistovat se stávajícím protokolem verze 4 a tím pádem neřešil problém tenčícího se počtu adres.

IP adresa je termín vyjadřující číselné označení síťového rozhraní vašeho počítače. Na rozdíl od reálného světa může mít jedno rozhraní více IP adres, ale jedna konkrétní adresa nesmí být použita více než jedenkrát. Obsah IP adresy tvoří 32 bitů, které jsou po každém oktetu odděleny tečkou. To nám dává více než 4 miliardy možných kombinací a tedy v ideálním případě použitelných IP adres. V reálném světě je toto číslo trochu nižší, protože některé adresy jsou zabrané pro speciální účely. To se původně (v 70. letech) zdálo jako dostatečné množství. Přidělování probíhalo jednoduchým způsobem a to tak, že každá organizace dostala blok IP adres, který umožňoval připojit 16777214 stanic, tedy množství odpovídající IP adrese třídy A. Ale to bylo ještě před zavedením třídy adres, takže maximální počet rozsahů adres byl 256 tedy pro 255 organizací. To samozřejmě přestalo již během pár let stačit a tak se rozsah IP adres rozdělil takzvanými „Třídami adres“. Jak je zobrazeno na obrázku 2, ty umožňovaly jemnější dělení a tím bylo možné rozdělit adresy více na „míru“ pro danou organizaci.



Obrázek 2: Třídy IP adres

Jak je vidět z obrázku, dělení bylo již o mnoho detailnější. Adresa třídy A umožňovala vytvořit 128 sítí a měla k tomu k dispozici prvních 8 bitů. Dalších 24 bitů umožňovalo určit 16777216 adres uzlů. Tato adresa byla

vhodná především pro velké organizace, ale ani ty nebyly schopné často tento potenciál využít.

Adresa třídy B již nabízela trochu reálnější rozsah adres uzlů, pro které bylo určeno posledních 16 bitů a to 65534. Pro adresu sítě bylo určeno prvních 16 bitů a to umožňovalo po odečteních prvních dvou bitů pro určení sítě vytvořit 16384 sítí.

Poslední třídou používanou pro připojení klientských stanic je adresa třídy C. Ta nabízela adresovat již přes 2 milióny sítí (použito prvních 24 bitů), ale na každé z nich jen 254 uzlů (posledních 8 bitů adresy) a to byl již i pro středně velké organizace malý rozsah.

Dále vedle tří základních ještě existují třídy D a E. Třída D (začínající čtveřicí bitů 1110) je určena pro tzv. skupinové vysílání (multicasting), a třída E (začínající pěticí bitů 11110) je vyhrazena pro budoucí využití.

Dle uvedených okolností bylo zapotřebí ještě jemnějšího dělení a to pomocí takzvané „Masky sítě“. Ta se nejčastěji uvádí za lomítkem, hned za IP adresou a vyjadřuje kolik bitů z celkového počtu 32 bude použito pro adresu sítě. Tento systém byl v roce 1993 ještě zdokonalen používáním CIDR (Classless Inter-Domain Routing), což v překladu znamená „Beztřídní mezidoménové směrování“ a to ukončilo dělení do tříd nadobro. CIDR umožňuje, aby byl předěl mezi adresou sítě a adresou uzlů umístěn libovolně bez ohledu na to, v jaké třídě se IP adresa nalézá.

Další opatření, které výrazně zpomalilo úbytek již docházejících adres, bylo nasazení NAT (Network Address Translation) technologie. Tento systém využívá privátních IPv4 adres (10.0.0.0/8, 172.16.0.0/12 a 192.168.0.0/16), které jsou v topologii sítě ze strany Internetu za hraničním směrovačem, v tomto případě za NAT. A díky tomu, do Internetu stačí jen jedna IP adresa a ve

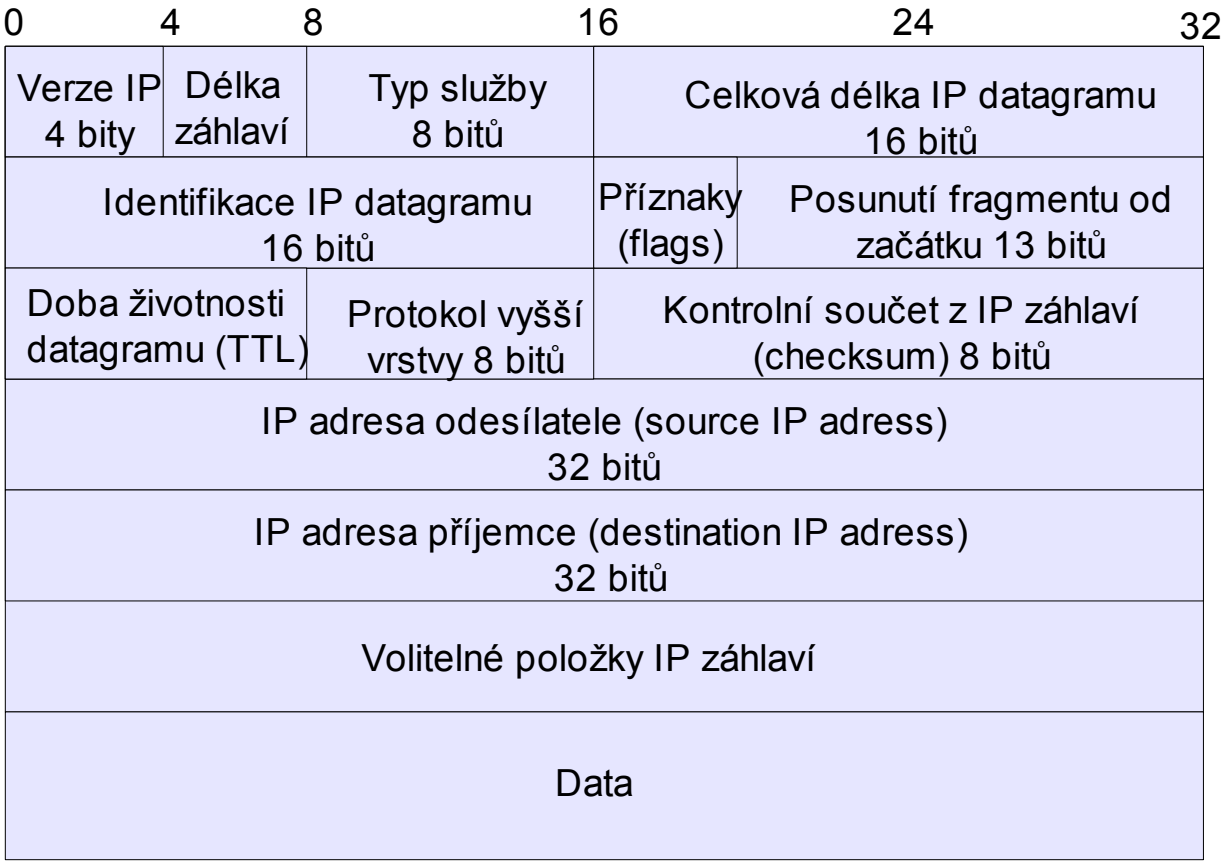
vnitřní síti může být velké množství počítačů s adresami z rozsahu privátních adres. U NAT není problém vytvářet spojení z vnitřní sítě ven, ale zvenčí dovnitř ano. Za to může skutečnost, že počítač z venku není viditelný a viditelná je pouze výchozí brána (NAT nejčastěji ve spojení s routerem) do Internetu. A tak vnější počítač neví oproti komu má dané spojení nasměrovat.

I přes všechna zde uvedená řešení, které vyčerpání IPv4 adres zdatně brání, termín kdy adresy skutečně dojdou, již není v nedohlednu. Pomocí výpočtů, které probíhaly na základě údajů čerpání adres v posledních 1000 dnech, byl předběžně stanoven rok 2012 jako datum, kdy k tomuto vyčerpání dojde. Ale i zde by mohly nastat ještě změny. Například tím, že si dodavatelé internetové konektivity budou chtít přivlastnit ještě zbývající adresy a tak může dojít k vyčerpání i v mnohem kratším čase. Otázkou je, jak se bude řešit okamžik, kdy k danému vyčerpání dojde. Bohužel, i v tomto případě hlavní roli hrají peníze, přeci jen implementace nového protokolu něco stojí a tak ze strany komerčních subjektů je odezva téměř nulová. Možností, jak se s tímto problémem vypořádat, je přejít na již připravený standard a to IPv6. Ten nabízí daleko větší adresní prostor (128 bitů pro IP adresu). Bohužel tato verze IP není s IPv4 zpětně kompatibilní a z důvodu velké decentralizovanosti Internetu samotného, není možné na nový protokol přejít ze dne na den. Ani přecházení postupné se nejeví jako uskutečnitelná varianta řešení daného problému. Tak jako jediná možnost se jeví postupné přecházení a vzájemná koexistence mezi IPv4 a Ipv6, kdy budou sítě ještě několik let využívat obou adresních technologií. A postupem času, odhadováno v období dlouhém 10 let, se zcela přejde na nový protokol.

2.6.1 Struktura IPv4 adresy

Jak už bylo uvedeno v předchozím textu, tak IP adresa verze 4 má velikost 32 bitů. Ta je součástí IP hlavičky, která má v běžném případě velikost

okolo 20B. Jak ukazuje obrázek 3, tak součástí hlavičky IPv4 je mnoho dalších parametrů.



Obrázek 3: Hlavička IPv4

Jako první v IP hlavičce se nachází pole o velikosti 4 bity, které specifikuje jaká verze IP protokolu je použita. V případě IPv4 tato nabývá hodnotu právě čtyř.

Další pole, také o velikosti 4 bity, udává, jak je velká celá IP hlavička. Nejčastěji, tedy i v našem případě, je velikost 20 bajtů, avšak může dosahovat velikosti až 60 bajtů a to podle počtu použitých volitelných položek.

Dále máme pole o velikosti 8 bitů, jenž by mělo označovat typ použité služby v datagramu. Bohužel, toto pole v dnešních sítích nemá příliš velké uplatnění. A tak se nastavení priorit paketů musí řešit jinými způsoby.

Položka Celková délka IP datagramu nám říká, jaké velikosti bude nabývat hlavička i s daty. Díky kapacitě pouze 16 bitů nám umožňuje vytvořit datagram o maximální velikosti 65535 bajtů.

Další položky identifikace IP datagramu, příznaky a posunutí fragmentu od začátku, mají dohromady velikost 32 bitů a slouží především pro účely fragmentace paketů.

Dalším parametrem, obsaženým v hlavičce, je doba životnosti, často uváděná zkratkou TTL (Time to Live), která má za účel zabránit nekonečnému přeposílání paketů. Funguje na principu snižování přednastavené hodnoty o 1 průchodem každého směrovacího prvku. Při vyčerpání je automaticky paket zahozen a je zpět vyslána ICMP zpráva o zahození tohoto paketu.

Pole „Protokol vyšší vrstvy“ o velikosti 8 bitů nám udává, jaký protokol využívá IP datagram ke svému transportu. V praxi se jedná nejčastěji o protokol TCP, UDP nebo některý z protokolů služebních ICMP nebo IGMP, které sice nepatří k protokolům vyšší vrstvy (jsou součástí IP protokolu), ale chovají se tak. To znamená, že za IP záhlaví přidávají ICMP nebo IGMP záhlaví, podobně jako například TCP.

Poslední položkou před samotnými IP adresami odesílatele a příjemce, je kontrolní součet z IP záhlaví. Bohužel, jak už napovídá samotný název, tak se kontroluje jen neporušenost IP záhlaví datagramu, které se ještě díky například TTL musí na každém směrovači přepočítávat. Což vyžaduje část režie a tak přispívá k ještě větším nárokům na daný směrovač a v extrémních případech až k výraznému zpoždění paketů.

Poslední položkou v IP hlavičce je pole označované jako volitelné položky IP záhlaví. Toto 32 bitové pole se využívá jen zřídka a jeho použití spíše způ-

sobuje problémy, díky nastavení některých směrovačů na automatické zahození paketu s tímto použitým polem.

2.7 Sítě IPv6

Když se podíváme do minulosti, tak IPv4 byla již na počátku devadesátých let na pokraji svých možností poskytnutí adresace počítačů ve světě. Proto se postupně zaváděly opatření, která měla tento termín co nejvíce odsunout, ale i tak bylo zřejmé, že se musí hledat nové řešení.

Proto se začal vyvíjet nový protokol, který měl nejen řešit problémy s adresací počítačů v Internetu, ale měl přinášet i nové možnosti, které by bylo možné do IPv4 implementovat jen opravdu těžko nebo vůbec. Dále měl opravit chyby, které už starší generace obsahovaly v době svého návrhu. Tak byl v roce 1994 přijat skupinou IETF (Internet Engineering Task Force) protokol, který měl všechny tyto požadavky splňovat. Z počátku byl označován jako IP Next Generation, ale později se přijalo několik RFC dokumentů, které ustanovily nový standard IPv6. Bohužel, jak už se v době vývoje očekávalo, tak přechod na nový protokol byl a stále je velmi pozvolný.

Pro nový protokol byla zvolena dostatečně obsáhlá IP adresa o velikosti 128 bitů, která poskytuje adresní prostor o velikosti zhruba $3{,}4 \times 10^{38}$ adres. Což se již zdá jako téměř nevyčerpatelný prostor. To do budoucna umožní, aby mělo svoji IP adresu již téměř každé elektrické zařízení (samozřejmě takové, u kterých by to mělo smysl). A každému takovému zařízení umožní mít pevnou IP adresu.

Určitě první a jednou z nejdůležitějších věcí kterou IPv6 vyřešilo, je adresní prostor. Další záležitostí, kterou naštěstí tento protokol překonává, je již nepotřebnost zařízení NAT. Toto zařízení velmi ztěžovalo přímou komunikaci mezi stanicemi uvnitř a mimo síť. U některých protokolů způsobovalo ab-

solutní nefunkčnost a nebo velmi složitou konfiguraci. Jistě, mezi další velmi příjemné vlastnosti, je povinná podpora bezpečnosti IPsec (u IPv4 možná ale nepovinná). Hlavička IPv6 definuje speciální typ podhlavičky, který je určen pro zpracování paketů IPsec. Další vlastností, která je součástí protokolu, je takzvaná Autokonfigurace. Ta dovoluje připojit počítač do sítě bez nutnosti konfigurace sítě v parametrech síťového adaptéru. V podstatě přijdete, zapojíte kabel a počítač si vyžádá prefix sítě, který má použít a zbytek adresy (64 bitů) získá z fyzické adresy vašeho síťového rozhraní. Podpora mobility je další nově přidaná vlastnost. Tato vlastnost sice byla už navržená pro IPv4, ale z hlediska malého adresního prostoru byla téměř nepoužitelná. V sítích IPv6 se tato vlastnost nazývá Mobile Ipv6 (MIPv6). Funguje na základě vzdálené výchozí brány (domácí agent). Tedy v podstatě, ať se připojujete z jakéhokoliv místa, je pro dané zařízení nějaké místo, které je označené jako domácí. V případě připojení do Internetu a následné komunikace je vytvořen tunel mezi mobilním zařízením, zařízením označeným jako domácí agent a vzdáleným serverem. Právě v tomto případě nastupuje síla Ipv6, kdy je celá komunikace ověřená a šifrována pomocí IPsec. Jenže skutečnost až tak dokonalá není a podpora mobility, asi jako jediné vylepšení, není příliš dobře implementováno a tak nezbývá, než čekat na lepší návrh, který již projde přes komisi IETF. Další vlastností, která byla značně vylepšena oproti starší verzi protokolu, je Quality Of Service (QoS). Ta již byla i ve starším protokolu, kde ale nenašla přílišného využití a jestli zde služba, která umožňuje nastavovat prioritu paketům, najde využití – to ukáže až čas (tím je myšleno větší rozšíření mezi veřejnost a množství zneužití daného nastavení). Jako poslední vylepšení bych uvedl důležité výrazné zrychlení směrování a to díky optimalizaci položek v hlavičce IP datagramu. Tím bylo dosaženo především odstranění položky Kontrolního součtu z IP záhlaví a tak každý směrovací prvek nemusí přepočítávat znovu kontrolní součet kvůli snížení

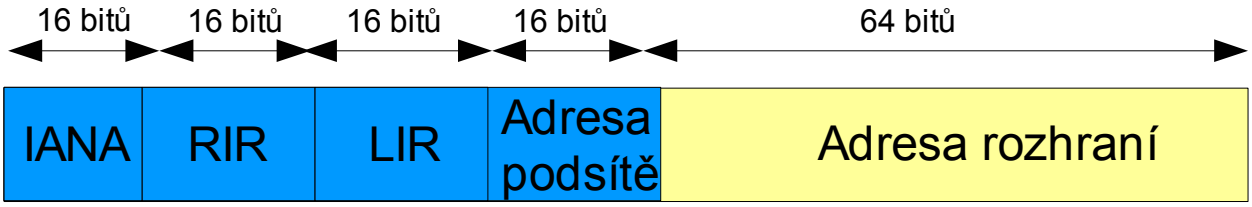
hodnoty TTL. Tímto jsem popsal nejdůležitější vylepšení oproti starší verzi protokolu.

2.7.1 Struktura IPv6 adresy

Jak už bylo výše uvedeno, adresa má velikost 128 bitů. Standardní zápis adresy je pomocí osmi skupin po čtyřech znacích 16tkové soustavy. Tyto skupiny jsou od sebe odděleny dvojtečkami. V případě, že se nalézají v jednom nebo po sobě jdoucích blocích samé nuly, nahrazují se tyto skupiny nul čtyřtečkou. To zajistí významné zkrácení IP adresy a tak zjednoduší její zápis.

V IPv6 existují tři typy adres a to individuální (unicast), skupinové (multicast) a výběrové (anycast). Individuální adresa je adresa přímo jednoho rozhraní na jednom zařízení. V podstatě jde o běžnou adresu v síti, která umožňuje odkazovat na konkrétní zařízení. Dalším typem adresy je adresa skupinová. Ta má za úkol adresovat vybranou skupinu uzlů. Data budou doručena každému z těchto uzlů, ale výhoda spočívá v tom, že každý uzel (jakékoliv zařízení se síťovým rozhraním) může patřit do libovolného množství multicast skupin. Posledním typem je adresa takzvaná výběrová (anycast). Tento druh adresy se objevuje poprvé a umožňuje označit skupinu uzlů, ale na rozdíl od multicast skupiny, se paket doručí jen jednomu uzlu ve skupině (tomu který je nejblíže).

Pro určení příslušnosti do určité sítě se používá takzvaný Prefix. Ten označuje kolik bitů je využito pro adresu sítě nebo podsítě. Záleží na tom, s jakou podrobností se na adresu díváte. Přidělování prefixů je v dnešní době shodné s přidělováním bloků adres IPv4. Toto je v péči organizace IANA (Internet Assigned Numbers Authority), která přiděluje bloky adres regionálním registrům označovaných jako RIR (Regional Internet Registry). Postup přidělování je zobrazen na obrázku 4.



Obrázek 4: Postup přidělování IPv6 adres

Hranice pro přidělování balíků adres regionálním registrům není pevná, zde si IANA může zvolit prefix až do velikosti 32 bitů. Dalším polem o velikosti 16 bitů, o které je navýšen prefix, přísluší takzvaným LIR (Local Internet Registry), což je v podstatě lokální registr. Tento registr zpravidla zastupují lokální poskytovatelé internetu, od nichž získávají adresy koncoví zákazníci. Poslední pole je nazvané Adresa rozhraní a jeho velikost je 64 bitů. To nám umožňuje v jedné podsíti adresovat téměř neuvěřitelných $18x10^{18}$ různých rozhraní. Tento poslední blok se vypočítává na základě fyzické adresy síťového adaptéru.

I přes značnou ochotu vývojářů vyřešit problém s adresováním již nadobro a tím pádem zvolení IP adresy o velikost 128 bitů, se hlavička, co do velikosti, příliš nezměnila. Dosaženo toho bylo pomocí značné optimalizace položek obsažených v samotné hlavičce. Původní hlavička typu IPv4 měla minimální velikost 20 bajtů a velikost samotné IP adresy byla 32 bitů. U adresy IPv6 je velikost základní hlavičky pevná a je stanovena na 40 bajtů (320 bajtů). Což při čtyřnásobné velikosti IP adresy nám dává pouze dvakrát větší hlavičku. Dále na obrázku obrázek 5 je zobrazena základní hlavička IPv6.

Obrázek 5: Hlavička IPv6

Stejně jako u staršího protokolu se na první pozici nalézá pole, které označuje verzi IP hlavičky, i zde má velikost 4 bity. Další pole se jmenuje Třída provozu. Velikost tohoto pole je osm bitů a mělo by sloužit k nastavování priorit pro služby, u kterých je důležitá například rychlá odezva, velká šířka pásma, stálá kvalita a jiné. Podobná služba byla implementována i do Ipv4, kde se jmenovala „Typ služby“. Bohužel ani v této (a ani v novější) verzi IP protokolu není její plná funkčnost. Přesto nám alespoň IP protokol dovoluje dosáhnout priorit pomocí Diferencovatelných služeb. Značka toku je další položkou v IP hlavičce. Velikost je určena na 20 bitů a bohužel podobně jako u předchozího pole, ani zde nejsou vlastnosti přesně definované. Proto funkčnost zatím není možné odzkoušet. Podstata tohoto pole by měla být v usnadnění a urychlení směrování. Měla by obsahovat hodnoty, které by označovaly datagramy s podobnými vlastnostmi. Například adresa příjemce / odesílatele a tím urychlit proud datagramů směrujících do jednoho cíle. Následující pole se nazývá Délka dat. Na první pohled by se mohlo zdát, že se

jedná o obdobu ze staršího protokolu Celková délka, ale v nové verzi má trochu jinou funkci. Zatím co ve starší verzi to byla velikost hlavičky a připojených dat, tak ve verzi nové se jedná o velikost dat za základní hlavičkou. Tím je možno pomocí rozšiřujících hlaviček dosáhnout většího datagramu (pojmenován jako Jumbo datagram) než 64KB, omezených pouze dvoubajtovou velikostí tohoto pole. Další pole s názvem Další hlavička obsahuje hodnotu, která označuje jaká hlavička pokračuje za hlavičkou základní. Toto zřetězení hlaviček nám umožňuje vložit za sebe velké množství rozšířených hlaviček a nebo zobrazit typ nesených dat za poslední hlavičkou. Pokud by množství těchto hlaviček bylo příliš velké a jejich uspořádání bylo náhodné, mohlo by dojít k velkému zpomalení na směrovacích zařízeních důsledkem procházení velkého množství dat. Proto bylo zavedeno pevné pořadí rozšiřujících hlaviček. Těchto rozšiřujících hlaviček je celkem osm.

2.7.2 Rozšiřující hlavičky

První hlavička, která může přijít po hlavičce základní, se nazývá Volby pro všechny (Hop-by-Hop Options header). Užitečnost této hlavičky vychází z toho, že je po cestě čtena každým směrovacím prvkem a tak je užitečná pro jejich správu nebo přenos informací. Další rozšiřující hlavička se jmenuje Volby pro cíl část 1 (Destination Options header note 1). V části rozšiřujících hlaviček se nachází na dvou místech. První je hned za Hop by Hop hlavičkou a pak je zpracovávána na cílovém počítači i na každém prvku v cestě, který její zpracování umožňuje. Druhým místem, kde se může vyskytnout, je za hlavičkou ESP (šifrování obsahu), pak je zpracována až na cílové stanici. Po této hlavičce se může vyskytnout hlavička Směrovaní (Routing header), která nám umožňuje mít kontrolu nad směrováním a tím dosáhnout určení přes které směrovače má daný paket procházet. To nám může v některých případech pomoci k odhalení problémů v síti a k určité optimalizaci toku paketů.

Další položka je označena fragmentace (Fragment header). Tato rozšířená hlavička umožňuje označit všechny fragmentované pakety. Postup fragmentace je odlišný oproti Ipv4, kde fragmentaci mohl provést jakýkoliv prvek po cestě paketu. V sítích IPv6 je fragmentace a defragmentace prováděna pouze na zdrojovém a cílovém počítači. K tomu, aby bylo toto možné, se nejdříve zjistí maximální MTU (Maximum Transfer Unit) přenosové trasy a teprve potom dojde k samotnému přenosu takto upravených dat. V dalších dvou hlavičkách je obsažené zabezpečení protokolu IPv6. Tyto hlavičky se nazývají Autentizace (Authentication header), zkratkou označováno jako „AH“ a Šifrování obsahu (Encapsulating Security Payload header) - označováno jako „ESP“. První z těchto rozšiřujících hlaviček (AH) slouží k ověření komunikujících stran a tím zamezení podvrhnutí identity připojujícího se uživatele. Druhá z hlaviček slouží k šifrování celé komunikace probíhající mezi dvěma účastníky. Obě tyto rozšiřující hlavičky jsou součástí protokolu IPsec, který bude dále dopodrobna rozebrán. Dále se už vyskytují jen rozšiřující hlavičky pro Volby pro cíl část 2, jejichž funkčnost byla již uvedena a hlavička pro Mobilitu (mobility), která bohužel nemá v dnešní implementaci příliš velké uplatnění. Po těchto rozšiřujících hlavičkách následuje hlavička, která je označována jako Vyšší vrstva (upper-layer header). Ta označuje protokol vyšší (transportní) vrstvy, který je použit pro přenos dat.

2.7.3 IPv6 a podpora ze strany vlády a výrobců Hardware a Software

Hned po vydání RFC k IPv6 byl projeven zájem ze strany výrobců o toto řešení jako jedno z možných náhrad za IPv4. Ze začátku, a to i díky neúplné implementaci tohoto standardu, nebylo možné plnou podporu, hlavně do hardwarových zařízení, implementovat. A tak jako první přichází v roce 1996 s podporou Ipv6 Linux (tehdy ještě ve vývojové verzi jádra). O pár let později se k Linuxu přidávají téměř všechny operační systémy vycházející z Unixu.

Tím byl nastartován přechod na nový IP protokol. Samozřejmě si těchto implementací všimli i samotní výrobci hardwaru a začali přidávat podporu pro IPv6 i do svých zařízení, jenže mnohdy nebyla úplná. Z tohoto důvodu se musela testovat kompatibilita a interoperabilita mezi jednotlivými implementacemi na různých zařízeních. Proto se začala používat loga z japonského programu TAHI, který měl dané testování na starosti. Tento program byl označován IPv6 Ready a obsahoval dvě třídy kompatibility, kterých mohl daný výrobek či software dosáhnout. Ve čtvrtém měsíci roku 2009 počet zařízení a programů splňujících druhou úroveň (vyšší) tedy plnou kompatibilitu byl 253.

Podpora pro aplikování IPv6 do sítě Internet ze strany vlády bohužel zatím není moc úspěšná (tedy alespoň u nás). Největší rozvoj této verze IP protokolu nastává v zemích, které byly ještě do nedávné doby nejvíce „internetové zaostalé“ a tak pociťují znatelný nedostatek IPv4 adres. Mezi tyto země patří dnes bouřlivě se rozvíjející se Čína, Indie a další země s velkým počtem obyvatel. I přes to první vláda s přímou podporou Ipv6, byla vláda Japonska. Nabídka různých daňových zvýhodnění značně urychlila danou implementaci nejen u firem. Spojené státy americké také nechtěly zůstat pozadu, začaly trochu z jiného konce a schválily tzv. směr IPv6. Ve výsledku měly všechny federální administrativní celky podporovat IPv6 dokonce roku 2008 a později také tomu přizpůsobit i všechny používané aplikace. Evropa také nabírá směr k používání tohoto protokolu, ale bohužel někdy až příliš komplikovaně a natolik krkolomně, že samotná implementace je téměř nemožná. Důkazem tohoto je plán vypracovaný Evropskou komisí z počátku roku 2008 pod názvem Action Plan for the deployment of Internet Protocol version 6 (IPv6) in Europe, který je k přečtení na stánkách Komise. Bohužel i to znamená další překážku pro rozvoj IPv6 v Evropě a tak nezbývá čekat na další a lepší návrh od Evropské komise.

2.8 Druhy zabezpečení na IP sítích

Bezpečnost v IP sítích je často diskutovanou a mnohdy velmi nákladnou záležitostí. To vychází jak z ceny použitého hardwaru, softwaru tak i cena za čas, který stráví síťový administrátoři nad kvalitním návrhem síťové topologie, výběrem příslušného softwaru, tak i jednotlivou konfigurací prvků v této síti obsažených. Cena takto navržené sítě velmi rychle roste s počtem uživatelů, kteří budou součástí této sítě. Dále náklady rostou s nároky na rychlost a hlavně s množství využívaných služeb v síti (FTP, Web server …). Služba v síti vytváří asi největší nároky na údržbu a to nejen z hlediska její implementace, tak i v případě jejího dodatečného zabezpečení a případně udržení její dlouhodobé funkčnosti. V takovýchto případech často přicházejí na řadu takzvané VPN (Virtual Private Network), které dokáží mnoho z těchto problémů vyřešit a koncentrují zabezpečení do jednoho velkého řešení. Základní požadavky pro možnost bezpečně vytvořit VPN jsou následující.

Neporušenost – tento pojem by měl určovat stav, kdy se data při průchodu sítí od zdrojového do cílového zařízení nezměnila.

Autentičnost – by měla vyjadřovat skutečnost, kdy obě strany spolu komunikující jsou ověřené. To znamená, že obě strany ví, s kým probíhá příslušná komunikace.

Uzavřenost – je vyžadováno již například při návrhu sítě „odstínit“ potencionálního útočníka od chráněných dat. V reálném světě to znamená zamezit komukoliv přístup k datům, pro které nemá případnou autorizaci a zabezpečit, aby k datům, ke kterým přístup má, neporozuměl. Toho se dosáhne pomocí jejich zašifrování.

Ochrana proti replay útokům – jedná se o jeden ze základních typů útoků, kdy navazující strana dostane žádost o znovu odeslání dat a to od

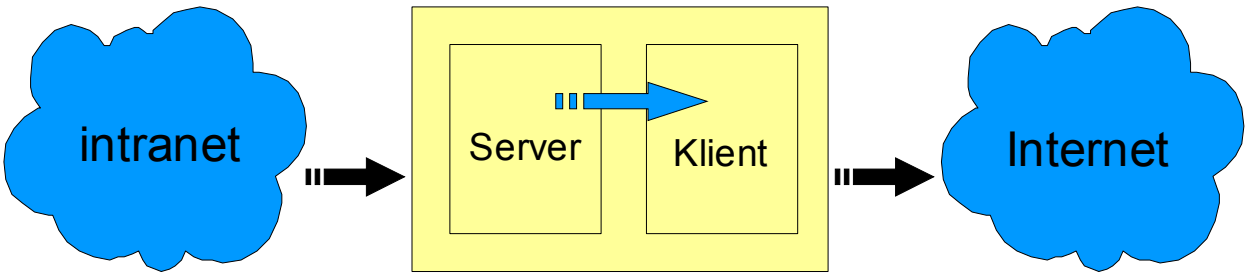
útočníka, který odposlechl předchozí komunikaci. A ten se tímto způsobem může dostat například k heslům, které klientská stanice použila pro přihlášení k firemnímu serveru.

Po splnění těchto základních požadavků může pokračovat v budování VPN. Termín VPN (Virtual Private Network) se používá pro označení takzvaných virtuálních sítí, které jsou vytvořeny na stávající topologii sítě. Technologie VPN pracuje na principu vytvoření tunelů mezi jednotlivými zařízeními nebo celými vnitřními sítěmi. Proto je důležité, aby byly tyto vnitřní sítě ochráněny ještě před vytvořením takového spojení. Toho můžeme dosáhnout následujícími metodami.

Jako první je metoda úplné izolace. Ta se doporučuje především u velmi citlivých dat, jako je například vojenský průmysl. Je to stav počítače nebo sítě počítačů, které nemají žádné propojení s ostatními sítěmi. Sice tato metoda patří mezi nejbezpečnější, bohužel v našem případě propojování vzdálených sítí postrádá smysl. Další možností je plné připojení do Internetu bez jakýchkoliv omezení. V případě využití této metody je celé zabezpečení řešeno až na každém samotném počítači a už z principu je velmi těžké takovou skupinu počítačů udržet s aktuálním režimem zabezpečení. Další metodou je propojení intranetu s Internetem skrz některý z bezpečnostních prvků. V tomto případě není vytvořená žádná konexe mezi intranetem a Internetem, která by před tím neprošla přes bezpečnostní prvek na prahu intranetu. Těmito prvky jsou ve většině filtrace, proxy a gateway, privátní sítě, firewall a nebo NAT. V mnoha případech se používá i jejich kombinace.

První způsob oddělení vnitřní sítě od Internetu pomocí filtrace se používá na přístupovém routeru, který má v sobě implementovaný tento prvek. Velmi často místo routeru se využívá server, který má dvě síťová rozhraní a jako operační systém je využita některá ze serverových edicích daného systému.

Ten nám pak umožní aplikovat další metody zabezpečení (proxy, wrapper atd.). Pomocí dané filtrace jde dosáhnout stavu, kdy uživatelé Intranetu mají téměř neomezený přístup do Internetu, ale opačně není přístup na servery vnitřní sítě možný. Takovéto odfiltrování je založeno na čtení údajů z hlaviček paketů na přístupovém routeru a dále rozhodnutím, jak s daným paketem bude naloženo. Tuto filtraci můžeme provádět na dvou úrovních a to buď podle IP hlavičky - pak se filtr nazývá Packet Filtr a nebo z TCP a UDP hlavičky - pak se daný filtr nazývá Circuit Filter. Další zmíněnou metodou je Proxy. Ta se aplikuje podobně jako filtr na rozhraní mezi vnitřní sítí a okolím pomocí serveru. Rozdíl oproti filtru spočívá v nutnosti se nejdřív k dané proxy přihlásit a teprve poté může být navázána komunikace s vnitřní či s vnější sítí. Jak ukazuje následující obrázek 6 je z pohledu připojujícího se klienta proxy server. Při pohledu serveru v Internetu se proxy stává klientem.



Obrázek 6: Komunikace se vzdáleným serverem s využitím Proxy

Další způsob, jak oddělit vnitřní síť od Internetu, je použitím Gateway. Ta funguje stejně jako proxy až na aplikační vrstvě, ale funguje na odlišném způsobu. Její funkce spočívá v převádění jednoho aplikačního protokolu na jiný, předem určený. Tím můžete docílit změny protokolu například z http na FTP a tím se připojit ke vzdálenému FTP serveru. Další způsob ochrany je pomocí Firewallu. Ten vesměs kombinuje všechny předchozí způsoby, plus k nim přidává některé věci navíc. Mezi rozšířené vlastnosti patří zařazení útočníka na černou listinu a tím mu znemožnit další navázání komunikace. Další přidanou možností je odpojit celou síť případně její části od Internetu a

mnoho dalšího. Ve větších sítích bývá implementován pomocí jednoho nebo skupiny počítačů, na kterých neběží již žádné jiné služby. Je to z důvodu vysoké náročnosti, která je na firewall kladena v čase velkých datových přenosů. Jak už bylo uvedeno, žádný z těchto způsobů se tak, jak byl popsán, nepoužívá. Ve většině případů se jedná o kombinaci několika výše jmenovaných druhů zabezpečení.

3 Tunelování

Pojem tunelování popisuje činnost, při níž se propojí dvě nebo více stran. Propojení proběhne jen virtuálně a díky tomu není zcela závislé na skutečné topologii sítě. Důvodem k vytváření tunelů je spojení sítí nebo jednotlivých síťových zařízení za účelem pro tunelování protokolu, který by za stávajících podmínek nemohl být použit a nebo za účelem zabezpečení spojení mezi těmito lokalitami. Nejčastější využití tunelování je propojení dvou vzdálených sítí a tím ve výsledku vytvoření jednoho síťového celku. Další časté použití je připojení uživatele do vzdálené sítě a tak mu umožnit chování, jako kdyby se v dané síti přímo nacházel. Ukončení takového tunelového spojení může být jak na přístupových routerech, tak až na počítači uživatele v intranetu. Tunelování má i své stinné stránky, jedna z nich je vyšší náročnost na síťové prvky, zejména na procesní kapacity směrovačů, kterými je tunel nakonfigurován. Také přichází s použitím tunelů složitější diagnostika vzniklých problémů s přenosem a zacyklením přenosu paketů v síti.

Jako základní parametr pro rozlišení typu tunelování je vrstva síťové architektury na níž je tunelování provedeno:

– tunelování na druhé (spojové) vrstvě – zde probíhá tunelování rámců a to může být vyžádané buď ze strany klienta - pak se jedná o voluntary tunneling a nebo ze strany přístupového serveru bez předchozího vyžádání klienta

(compulsory tunneling). Specifikací tunelování na spojové vrstvě je nutnost, aby samotný protokol požadované spojení vytvořil, udržoval a také ukončil,

– tunelování na třetí (síťové) vrstvě – aby mohl být tento typ tunelování proveditelný, tak musí být původní IP datagram zapouzdřen do jiného datagramu (IP v IP nebo IPv6 použitá v Ipv4).

3.1 Vytváření VPN pomocí tunelování

Sítě typu VPN (Virtual private Network) se začaly používat zejména z důvodu vytvoření nových virtuálních sítí na stávající kabeláži. To v praxi umožňovalo mít v jedné síti několik privátních sítí bez nutnosti je oddělit fyzicky. Samotné vytváření VPN je založeno na tunelování mezi jednotlivými prvky sítě, pro které přináší již zabezpečené spojení. Podstatou VPN je, aby paket přenášený skrz tunel byl nečitelný pro transportní síť a tak jeho obsah byl odhalen jen cílovému zařízení.

Podobně jako tomu je u tunelování, tak i VPN můžeme rozdělit ze dvou hledisek a to podle typu spojení mezi koncovými stanicemi a podle vrstvy, na které je daná virtuální privátní síť vybudována.

Tedy podle typu spojení můžeme rozdělit VPN následovně:

– vzdálený přístup (remote access) – tento typ slouží pro připojení uživatele například do podnikového intranetu. Požadavky na zabezpečení a zejména na autentizaci uživatele jsou větší než u jiných typů připojení. Jako přístupové zařízení do vnitřní LAN bývá nejčastěji použit Firewall nebo Proxy server. Služby s přidělením vnitřní IP adresy (DHCP) a překladem jmen (DNS) zde obstarává to samé zařízení, které slouží pro přístup vzdáleného uživatele.

– Propojení vzdálených intranetů – typ spojení, který se používá v těchto případech, je označován jako Site to Site nebo Lan to Lan. Využití spojení Lan to

Lan je k utvoření jednoho celku z dvou nebo více vzdálených intranetů. Největší bezpečnostní riziko se nalézá při navazování spojení a to v ověření, zda se jedná o daný intranet. V případě, že by došlo k podvržení těchto údajů, útočník by získal přístup do celé sítě a ke všem datům v ní distribuovaných.

– Spojení extranetů – zde je VPN užita k propojení dvou intranetů, které nemusí patřit jedné organizaci, ale například obchodním partnerům a tím jim umožnit sdílet mezi sebou požadovaná data.

Podle vrstvy na které je VPN vybudována je rozdělení následující:

– na síťové vrstvě – v rámci síťové vrstvy probíhá směrování IP protokolu a spolu s informacemi o směrování je utvořen základ pro vytvoření VPN na síťové vrstvě. Nejčastěji využívané metody jak vytvořit síť tohoto druhu jsou filtrování směrovacích informací, tunelováním a nebo šifrováním na síťové vrstvě.

– na spojové vrstvě – metoda vytváření VPN na spojové vrstvě nám umožňuje nejpřímější vybudování těchto sítí. Mezi další výhody patří možnost vybudovat další nezávislé VPN na síťové vrstvě a tím se co nejvíce přiblížit modelu konvenčních privátních datových sítí. Zde je asi nejzajímavější z hlediska zaměření bakalářské práce MPLS.

– na transportní a aplikační vrstvě – tento typ VPN není příliš rozšířený. Nejčastější použití tohoto způsobu je doprovázeno protokolem SSL (Secure Socket Layer), respektive jeho novější verzí 3.1, která je označována jako TLS (Transport Layer Security). Zde se provádí šifrování jen u aplikace, která má SSL implementované.

4 Virtual Private Network

4.1 Na síťové vrstvě

Filtrování směrovacích informací je jedna z metod vytvoření VPN na síťové vrstvě. Princip fungování je velmi jednoduchý a spočívá v potlačení směrovacích informací o okolních sítích. Okolní sítě jsou v tomto případě sítě, které nepatří do VPN. Když tedy dojde k počítači paket od jiného zařízení než náleží dané VPN, počítač, díky neznalosti jak dosáhnout jiné sítě než privátní, na tento paket nemůže odpovědět. Bohužel, tento způsob je velmi náročný na konfiguraci. Musí se například zabránit přístupu na jiný implicitní směrovač, který slouží pro komunikaci se sítěmi mimo VPN, který musí být také přístupný.

Tunely jsou další možností pro vytvoření VPN na síťové vrstvě. Je to vynikající metoda pro vytváření tunelů, ale bohužel i zde narážíme na překážky a to zejména při spojení typu bod – více bodů. V tomto případě roste problém se škálovatelností extrémně oproti spojení typu bod – více bodů s více spojeními, kde roste složitost jen lineárně. Jako výhodu tunelování oproti metodě filtrování směrových informací lze považovat adresaci různou jak pro VPN, tak i pro síť mezi přístupovými body jednotlivých VPN. Tím dosáhneme odstínění směrování i adresních prostorů uvnitř a vně sítě.

Tunelováni pomocí GRE (Generic Routing Encapsulation) je základní typ tunelovacího protokolu. Cíl při vývoji byl přenášet pakety jednoho typu protokolu skrz jiný protokol. Před průchodem paketu tunelem je obalen GRE hlavičkou. Tato hlavička je odebrána až na konci tunelu. Od tohoto místa pokračuje paket už bez této hlavičky. Jak vypadá paket při průchodu GRE tunelem je zobrazen na obrázku 7.

Obrázek 7: Paket při průchodu GRE tunelem

GRE tunely umožňují pouze spojení typu bod – bod a nehodí se pro jiný typ propojení, i když některé komerční varianty tento způsob umožňují.

Mezi další tunelovací protokoly patří PPTP (Point to Point Tunneling Protocol) a L2TP (Layer 2 Tunneling Protocol). Obě tyto metody pro vytváření tunelových spojení jsou využívány sítěmi s komutovaným přístupem. Novější protokol L2TP je odvozen z dřívěji používaného L2F a ze specifikace PPTP. Z hlediska rozšíření je na tom lépe protokol PPTP a to díky širší podpoře ze strany operačních systémů. Standart L2TP je typem tunelování, kdy spojení je iniciované ze strany přístupového serveru a tak nemá připojující se uživatel možnost příchozí spojení ovlivnit. Po proběhnutí autentizace pomocí zabezpečujícího serveru je dynamicky vytvořen L2TP tunel. Na rozdíl tunel PPTP je vyžádán ze strany klienta, který chce přístup do VPN. Tento způsob připojení umožňuje klientovi ovlivnit parametry spojení a daný tunel vytvořit i bez iniciativy přístupového serveru. Nejdříve je vytvořeno PPP spojení na přístupový server, kde je ukončeno standardním způsobem PPP. Dále na základě přístupových práv je sestaven klientem PPTP tunel. Tento způsob je vhodný, pokud se cílový uzel často mění. Mezi výhody patří transparentnost tunelu z pohledu poskytovatele a tím možnost procházet i skrze více sítí.

4.2 Na spojové vrstvě

Na první vrstvě TCP/IP modelu, nazývané spojová vrstva, se budují VPN, které se považují za nejpříměji vybudované virtuální sítě. Na těchto sítích je dále možné budovat další nezávislé VPN na vyšší přenosové vrstvě. Princip

fungování virtuálních linek na spojové vrstvě je velmi kvalitní alternativa k sítím, kde jsou použity pevné dedikované obvody. Díky dalším použitým technologiím je možné dosáhnout například emulovaných pevných linek s konstantní přenosovou rychlostí a garantovaným maximálním zpožděním.

Virtuální síť dosažená na spojové vrstvě pomocí Lan emulace označované zkratkou LANE, je jeden z typů, který je možno při budování těchto sítí využít. Původně byl tento princip vyvinut v prostředí Ethernetových přepínačů, ale nic nebrání jeho využití v sítích ATM (Asynchronous Transfer Mode). V praxi je vytvořena (emulována) Lan nad sítí ATM. LANE protokol definuje rozhraní, které je použito pro vyšší vrstvy. Pak jsou pakety vyšších síťových protokolů zapouzdřeny do jednoho ze dvou možných LANE MAC rámců a odeslány skrz ATM síť. V praxi pak vypadá daná síť jako klasický Ethernet a nebo Token Ring, jen jen značně rychlejší.

Další variantou, jak vytvořit VPN nad ATM je použití technologie MPOA (Multiprotocol over ATM). Ještě v minulém desetiletí se do tohoto standardu vkládaly velké naděje, měl totiž jako první přinést směrování při současném použití ATM technologie. Měl tedy umožnit snadnější spolupráci se sítěmi jiného typu. Hlavní rozdíl oproti sítím budovaným pomocí LANE přístupu, je zavedení pojmu virtuální směrovač. Ten zaručuje sítím velkou škálovatelnost a možnou flexibilitu řešení. Virtuální směrovač emuluje standardní fyzické sítě a zároveň překonává jejich nedostatek, který vyplýval z jejich principu směrování. Ten měl za následek vysoké zatížení směrovacích prvků, kde musel být každý paket při průchodu výpočetně zpracován. V takových případech MPOA identifikuje datový tok a nasměruje ho do daného virtuálního spojení. Dále ustanoví přímé spojení skrz ATM síť a dosáhne tzv. "zero-hop" routingu. Největší nevýhodou MPOA je možnost použití pouze na sítích ATM. Díky této vlastnosti je použití omezeno a z pohledu velkých hybridních síti je značně limitující.

Poslední využívanou technologií, která by měla odstraňovat, případně alespoň minimalizovat nevýhody dvou předchozích řešení, je MPLS (Multiprotocol Label Swapping). Daná technologie je označována jako hybridní a to díky přístupu k tvorbě VPN. Základní dva přístupy spočívají v použití směrování na síťové vrstvy spolu s přepínáním paketu po paketu a virtuální obvody na spojové vrstvě spojené s přepínáním podle datových toků. MPLS není vázaná na žádnou konkrétní technologii a díky tomu může pracovat s libovolným přenosovým médiem, kde jsou jednotlivé uzly označeny IP adresami, a síť umožňuje paketový přenos.

4.3 Na transportní a aplikační vrstvě

Na těchto vrstvách se ve většině případů zabezpečení přenášených dat řeší již pro konkrétní aplikaci, a ne pro všechny odchozí data na danou IP adresu. Nejčastěji využívané zabezpečené protokoly jsou SSH (Secure Shell) a SSL (Secure Sockets Layer).

SSH bylo navrženo jako zabezpečená náhrada za telnet, který slouží k připojení ke vzdálené stanici. Pomocí SSH je možné propojit dva vzdálené počítače pro jejich případnou komunikaci (připojení k příkazovému řádku), kopírování dat či přenos námi požadovaných informací. Protokol má implementováno vše, co je potřeba pro autentizaci účastníků komunikace, šifrování samotných přenášených dat a kontrolu jejich integrity, dále lze ještě použít bezeztrátovou kompresi. Daemon naslouchá na portu 22, na který se připojují vzdálení klienti. Implementace SSH protokolu je dnes velmi rozšířená, asi nejznámější variantou je OpenSSH.

Tato varianta je volně šiřitelná, bohužel neobsahuje port na mnoho operačních systémů. V dnešní době se používá již verze SSH-2, která byla přijatá za internetový standard v roce 2006 skupinou IETF. Druhá verze SSH ob-

sahuje již bezpečnou výměnu klíčů Diffie-Hellman algoritmu a stává se tak více bezpečnou. Architektura ve druhé verzi je taktéž na vyšší úrovni a je rozdělena do třech vrstev. Transportní vrstva má za úkol počáteční výměnu klíčů, případně jejich obměnu po stanoveném čase či přenesených datech. Další vrstva se nazývá autentizace uživatele. Řídící člen při zahájení komunikace je klient. Zde je na výběr hned několik autentizačních variant s různým rozdílem bezpečnosti. Bohužel, některé metody nejsou standardně obsaženy ve všech implementacích a tak není možné jejich využití. Poslední vrstvou je vrstva spojení. Ta obsahuje koncept kanálů a jejich požadavky, skrz které jsou služby SSH poskytovány. V dnešní době je SSH běžně používané například pro vzdálenou práci a nebo správu vzdálených počítačů. Popularnost se stále zvětšuje díky přenosu na mnoho operačních systémů v mnoha variantách a velmi jednoduchému a účinnému použití.

Mezi další řešení zabezpečení přenášených dat pomocí tunelů na transportní vrstvě můžeme uvést protokol SSL. Místo v TCP/IP modelu není přímo na transportní vrstvě, ale mezi ní a vrstvou aplikační. SSL podobně jako předchozí protokol obsahuje mechanizmy pro bezpečnou autentizaci a šifrování přenášených dat. Na nejčastější využití protokolu můžeme narazit při komunikaci s webovými servery pomocí HTTP a po navázání zabezpečeného spojení HTTPS. I zde se pro výměnu klíčů využívá nejčastěji Diffie-Hellman algoritmus a jako komunikační port je označen 443 (`HTTPS/SSL`).

Technologií pro vytváření virtuálních privátních sítí je celá řada, některé z nich zde ani nebyly zmíněny z důvodu malého rozšíření či zastaralosti oproti novějším řešením. Dále jsou uvedeny tabulky 1 a 2, které jednoduše shrnují, jaký vzdálený přístup do sítě nám daná technologie nabízí, která je její pracovní vrstva a jaké mechanizmy nabízí pro zabezpečení přenášených dat.

Tabulka 1: Popis technologií pro tvorbu VPN. Zdroj: Vlastní

Typ technologie	Vrstva	Typ přístupu do sítě	
		Site to site	Vzdálený přístup
MPLS	Spojová i síťová	ano	ne
MPOA	Spojová	ano	ne
LANE	Spojová	ano	ne
PPTP	Spojová	ne	ano
L2TP	Spojová	ne	ano
IPSEC	Síťová	ano	ano
GRE	Síťová	ne	ano
SSL/TLS	Aplikační	ne	ano
SSH-v2	Aplikační	ne	ano

Tabulka 2: Typ autentizace a přenosu dat. Zdroj: Vlastní

Typ technologie	Zabezpečení autentizace a přenosu dat			
	Autentizace	Šifrování	Algoritmy	Generování klíče
MPLS	žádná	žádné	žádné	žádné
MPOA	žádná	žádné	žádné	žádné
LANE	žádná	žádné	žádné	žádné
PPTP	MSCHAP-v2, EAP-TLS	MPPE(nepovinné)	RSA, RC4	LM (DES),NTLMv2 (MD4)
L2TP	IP-sec	IP-sec	DES, 3DES. AES	náhodný klíč, Diffie-Hellman group 1, 2, 14
IPSEC				
GRE	žádná	žádná	žádná	žádná
SSL/TLS	Diffie-Hellman, RSA, DSA	SSL	RC2, RC4, IDEA, DES, Triple DES, AES, Camellia	Diffie-Hellman group 1, 14
SSH-v2		SSH	AES, Blowfish, 3DES, CAST128, Arcfour	

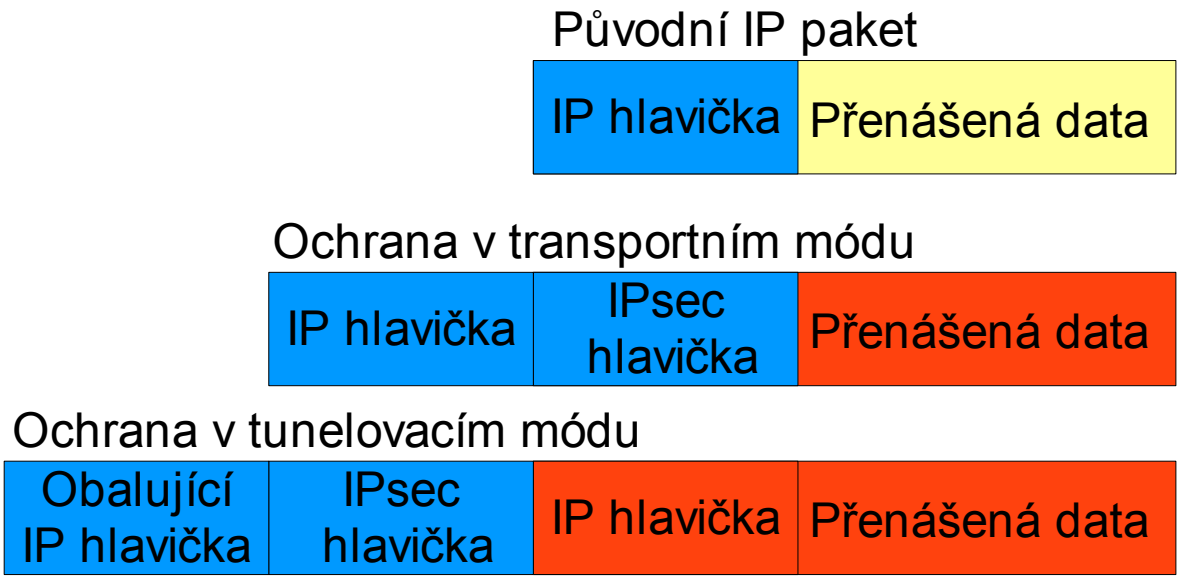
4.4 VPN pomocí IP-sec technologie

I přes to, že byl původně protokol IP vyvíjen pro armádu, neobsahoval ve své implementaci žádný druh zabezpečení, ať přenášených dat, tak ověření komunikujících stran. Samotný název IP-sec vznikl z jeho principu a to tím, že zabezpečuje přenos na IP vrstvě. Jeho použití je možné jak na protokolu Ipv4, tak na novějším protokolu IPv6, kde je jeho použití povinné. Bohužel v řadě implementací chybí a nebo je jen částečná. Hlavními prvky, které IP-sec poskytuje, jsou šifrování přenášených dat a autentizaci komunikujících stran spojení. Dnes se protokol nachází již ve třetí verzi, jež je z roku 2005. Bohu-

žel, ani tato verze není finální verzí a spíše jen vylepšuje verze předchozí. Všechny mechanizmy jsou u IPsec navrženy tak, aby byla zajištěna interoperabilita mezi různými implementacemi a zároveň bylo umožněno přidat nový algoritmus bez jejího porušení.

Ochrana IP-datagramu je zajištěna pomocí rozšiřujících hlaviček a to buď AH (Authentication Header) a ESP (Encapsulating Security Payload) a nebo je možné použít pouze hlavičku ESP. Hlavička starající se o autentizaci komunikujících stran (AH) dále zajišťuje integritu posílaných dat a ochranu proti replay útokům. Hlavička ESP umí téměř vše co AH ale má pár rozšíření navíc. Z výše zmíněného důvodu se do budoucna již s AH nepočítá a celá komunikace bude opatřena pouze rozšiřující hlavičkou ESP. Zatím je doporučeno obě rozšiřující hlavičky využívat a to hlavně z důvodů zajištění interoperability mezi jednotlivými implementacemi. Bezpečnostní služby, které IPsec zajišťuje, jsou založeny na principu sdíleného klíče. Pomocí sdíleného klíče jsou komunikující strany ověřeny a zároveň šifrována data. Tento klíč je možné zadat manuálně a nebo nechat dynamicky ověřit, vyjednat zásady zabezpečení a vygenerovat sdílený klíč. Na takovém základu vygenerovaný klíč se nazývá IKE (Internet Key Exchange).

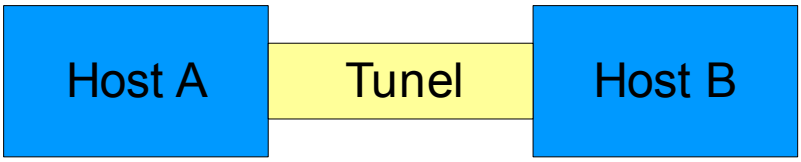
Architektura protokolu IPsec je popsána v RFC2401, kde jsou zmíněny základní principy od nichž se odvíjejí všechny implementace. Při jeho použití můžeme buď ochránit všechna nesená data (IP payload) a nebo můžeme zabalit data i s původní hlavičkou a celé obalit hlavičkou novou. Tím nám IPsec umožňuje pracovat v režimu Tunelujícím a Transportním. Základní rozdíl je znázorněn na obrázku 8, kde jsou chráněná data zobrazena červenou barvou.



Obrázek 8: Transportní a tunelovací mód

4.4.1 Provozní módy IPsec

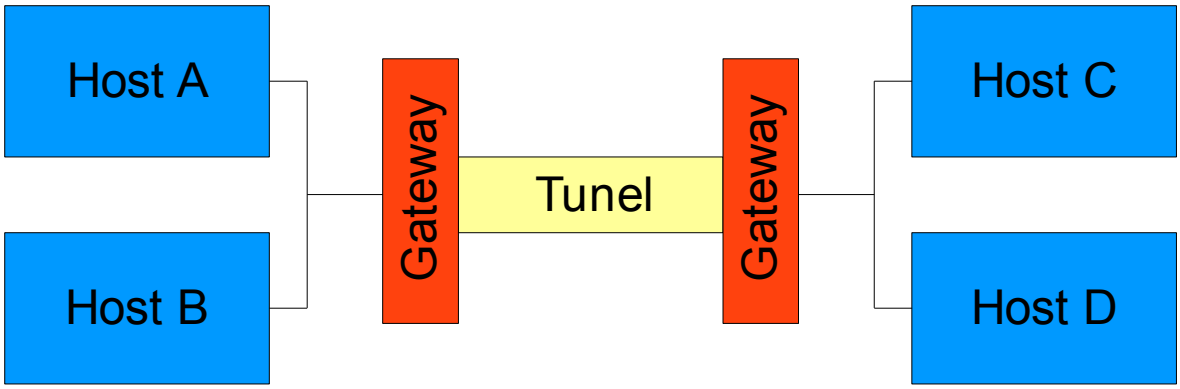
Transportní mód se používá k propojení dvou vzdálených stanic zabezpečeným kanálem obrázek 9. Při jeho použití se pomocí IPsec zašifrují jen přenášená data a to má za následek, že je stále možno odhalit kdo s kým komunikuje, protože v IP hlavičce budou stále obsaženy údaje o odesílateli a příjemci daného paketu. Ač to na první pohled není jasné i tato skutečnost může být v některých případech velmi nepříjemná.



Obrázek 9: Transportní mód

Pokud chceme tento nedostatek obejít, použijeme mód tunelovací zobrazený na obrázku 10. Režim tunelování se nejčastěji používá k propojení dvou vzdálených internetových bran. Následek takového propojení je komunikace mezi bránami probíhající v zašifrované podobě. Při takovém propojení jsou přenášená data i původní hlavička obaleny IPsec hlavičkou, která

vše zašifruje a celý paket je obalen novou IP hlavičkou. Tímto způsobem upravený paket nese již ve své IP hlavičce adresu prvku, na kterém byl upraven a cílová adresa je adresa protějšího prvku nejčastěji vzdálené gateway.



Obrázek 10: Tunelový mód

Při této modelové situaci tedy komunikace probíhá takto:

– Uživatel (Host A) odesílá data do vzdálené pobočky firmy (Host D),

– data putují interní sítí v nezašifrované podobě,

– když dorazí na hraniční směrovač (Gateway), tak je podle cílové adresy vyhodnoceno využití šifrovaného kanálu k přenosu,

– data jsou obalena IPsec hlavičkou a poslána tunelem skrz veřejnou síť již s IP adresou hraničního směrovače a cílovou protější brány,

– při průchodu sítí nelze zjistit kdo s kým komunikuje a co přenáší, je pouze možné identifikovat, že proběhla komunikace mezi dvěma gateway,

– data, která dorazí k bráně do intranetu vzdálené firemní pobočky jsou podle IP adresy identifikovány a následně rozšifrovány pomocí sdíleného klíče,

– a v nezašifrované podobě již putují intranetem k příjemci (Host D).

Tento způsob přenosu má velké výhody, zejména v tom, že účastníci komunikace se ani nemusí dozvědět, že jejich data putovala nějakým zabezpe-

čeným kanálem. Dále konfigurace takového kanálu není na každém klientském počítači. Z toho důvodu je možné dosáhnout ještě kvalitnějšího zabezpečení při minimálním množství administrace.

4.5 SA (Security Association)

Důležitou funkcí, která je obsažená v IPsec, je takzvaná SA (Security Association). Ta má za úkol rozhodnout o tom, jaká data chránit, jak je chránit a jak s těmito daty naložit. U IPsec je SA pouze jednosměrná, což znamená, že pro plný duplexní přenos musí být vytvořeny dvě. Výhoda spočívá například v možnosti v každém směru požívat jiný šifrovací klíč. Asociace tohoto typu mohou být vytvořeny jak manuálně, tak dynamicky. Pro dynamické vytvoření se využívá protokol IKE. Po té, co jsou vytvořeny, se ukládají do SPD (Security Policy Database). Dále v SPD jsou jednotlivé asociace označovány indexem bezpečnostních parametrů SPI (Security Parameters Index). Index SPI je nesen v každé IPsec hlavičce spolu s cílovou adresou na kterou se tato politika vztahuje.

Správná funkčnost a nastavení SPD je velmi důležitá pro smysluplné používání IPsec a zabezpečení v síti obecně. Proto je SPD velmi striktně nastavena podle těchto pravidel:

– Zahodit datagram (přichází v platnost, pokud se odesílaný nebo přijímaný paket neshoduje s žádnou politikou v SPD a dále se nachází v některém rozsahu jednoho ze selektorů),

– zpracovat datagram (následuje v případě nenalezení shody v SPD, ale není omezený žádným ze selektorů, na takovýto datagram není aplikován IPsec,

– aplikovat IPsec na datagram (přichází na řadu v případě nalezení shody v SPD).

Jako selektor v těchto případech může být použit:

– Cílové zakázané IP adresy,

– zdrojové zakázané IP adresy,

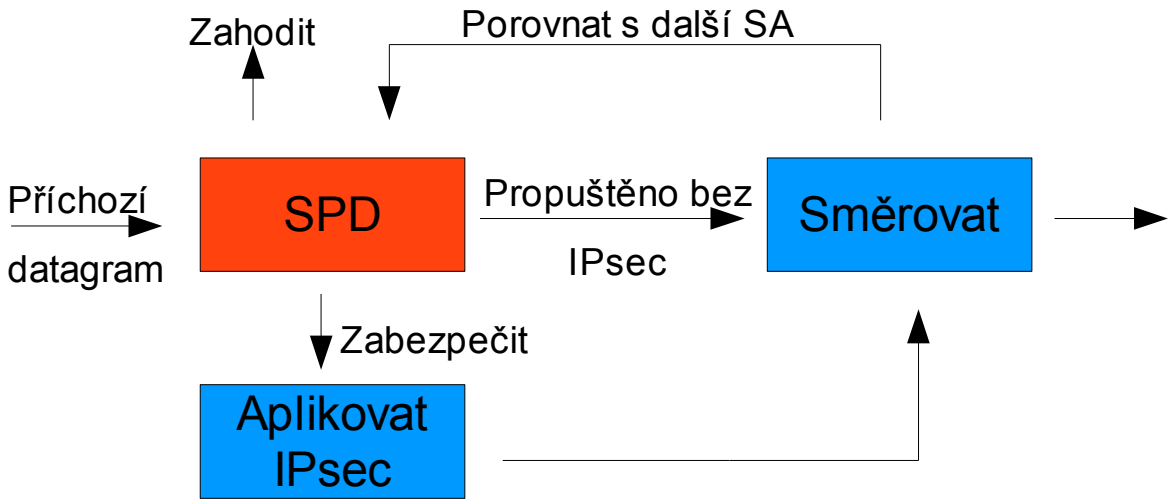
– bezpečnostní pole,

– protokol transportní vrstvy,

– cílový port,

– zdrojový port.

Při výběru politiky, která má být na datagram uplatněna, se postupuje způsobem první nalezený záznam a podle něj je zpracován. Celý proces odesílání datagramu je zobrazen na obrázek 11.



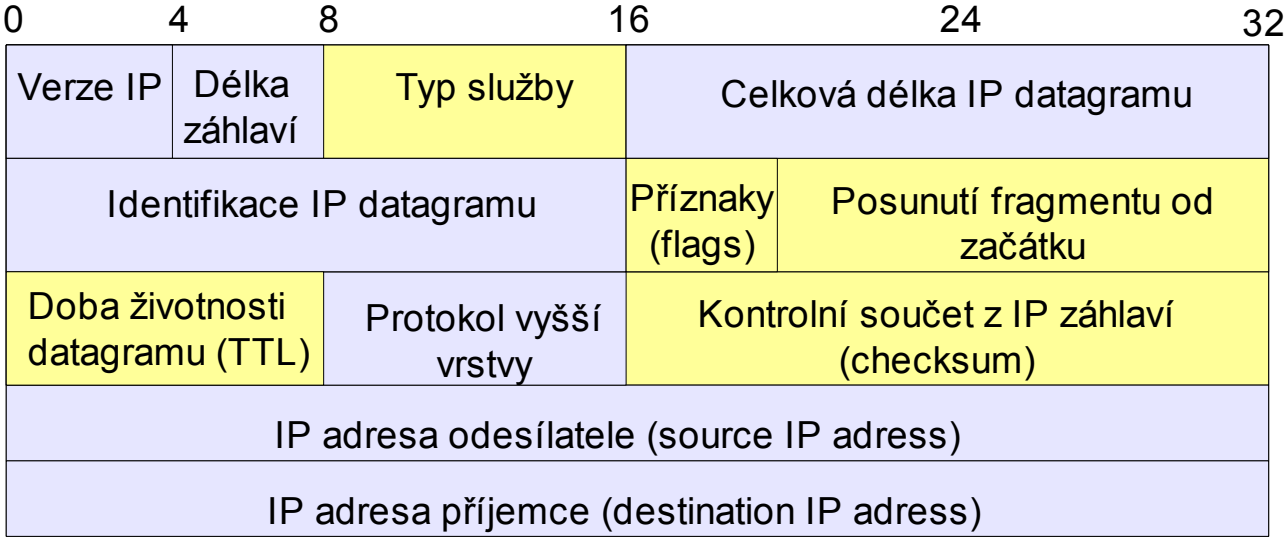
Obrázek 11: Vnitřní mechanizmus SPD při odesílání datagramu

Zpracování příchozího datagramu je velmi podobné, zde pouze nastává rozdíl v případě, když dorazí ve formě fragmentů, tak se musí nejdříve složit a dále je již celý postup reverzně provedený, stejný.

4.6 AH (Authentication Header)

Při přenosu dat není zapotřebí pokaždé přenášené informace šifrovat, je třeba pouze ověřit totožnost odesílatele a příjemce. Ve chvíli takovéhoto požadavku přichází možnost použít AH. Její funkcí je zajistit integritu přenášených dat a autorizovat komunikující strany. Při využití této rozšiřující hlavičky máme také možnost využít jak transportního, tak tunelovacího módu. Dále nám nabízí ochranu proti replay útokům a je možné ji použít podobně třeba jako hlavičku ESP, nad tunelovacími protokoly GRE nebo L2TP a další. V případě, že je zapotřebí i šifrování, není problém ji použít najednou i s ESP hlavičkou. I přes to, že ESP nabízí všechny možnosti, co AH, je při současném použití obou hlaviček, autentizace a integrita dat zajištěna pomocí AH.

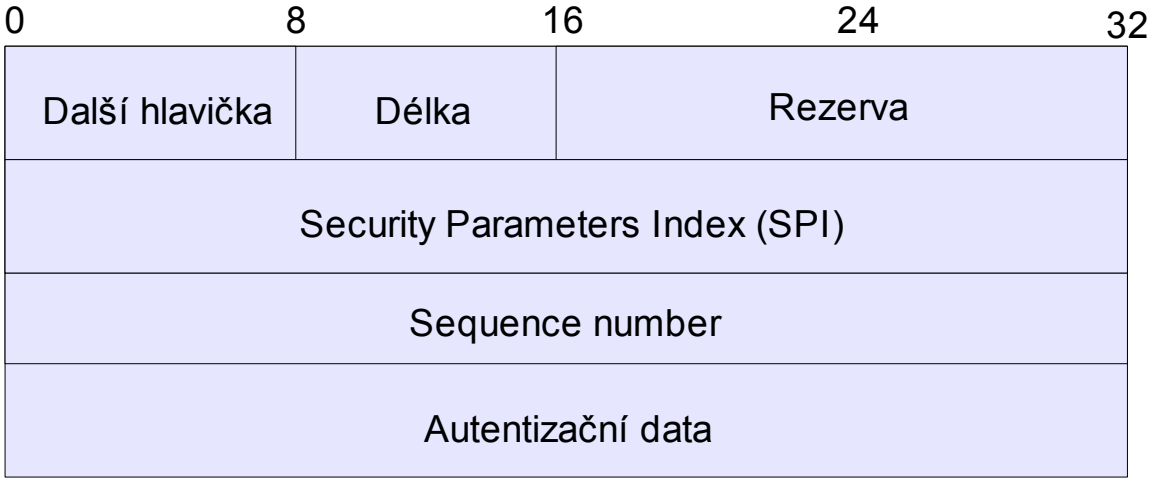
Aby bylo možné zajistit integritu dat, AH provádí zabezpečení pomocí MAC klíče. Tento klíč se vypočítá z části IP hlavičky a nesených dat a dále je nazýván ICV (integrity check value). Na obrázku 12 jsou žlutě zobrazena ta pole, jejichž pole nelze do ICV zahrnou z důvodu změny hodnot na každém aktivním prvku sítě.



Obrázek 12: Hlavička IPv4 pro výpočet ICV

Před výpočtem se žluté pole inicializují na hodnotu nula, aby neovlivňovaly ICV po dosažení cíle, kde se integrita ověří stejným způsobem, jako při jeho výpočtu. Některé autentizační algoritmy vyžadují, aby velikost bloku byla jejich násobkem a proto je potřeba doplnit daty s hodnotou nula. Ty pak neovlivňují výpočet a nejsou ani zahrnuty do pole udávající velikost nesených dat. AH záhlaví také musí být tímto způsobem upraveno a to na násobek 32 bitů v případě použití IPv4 a 64 bitů pro IPv6. Po provedení všech těchto úprav již může být ICV vypočítáno a IP pakety zabezpečeně přenášeny.

Další obrázek 13 zobrazuje obsah autentizační hlavičky.



Obrázek 13: AH (Authentication Header)

Pole další hlavička zde obsahuje kódovou hodnotu a informuje o tom, která hlavička bude následovat. Zde se může například objevit v tunelovacím režimu hodnota 4 pro IP to IP zapouzdření v IPv4 a nebo pro IPv6 hodnota 41. Následující pole délka popisuje délku samotné AH hlavičky a to v upravené podobě. Hodnota se vypočítá jako násobek 32 bitových slov obsažených v hlavičce mínus dvě. Další užitečné pole je označeno SPI a to slouží za pomocí cílové IP adresy k identifikaci SA. Zde se mohou objevit hodnoty 0 to označuje lokální použití a tomto případě není datagram puštěn dál.

Pro běžné využití se užívají hodnoty 256 až $2^{32} - 1$. Pole Sequence number slouží k číslování paketů. Číslování probíhá zvýšením hodnoty vždy jen o jedna. S tím spojená startovní hodnota je také jedna. Díky této funkci může AH odolat replay útokům. Bohužel, použití není vyžadováno a závisí na každém příjemci, zda ho využije a bude aplikovat jako bezpečnostní politiku (pro odesílatele povinná položka). Poslední položka, nazvaná autentizační data, slouží k uchování hodnoty po výpočtu ICV. Pole je vždy násobkem 32 bitového slova, v případě, že není vyplněno, musí se dorovnat na požadovanou velikost.

Pokud AH za pomocí SA naváže první konexi, obvykle pomocí IKE a autentizační algoritmus spolu s klíči je uložen. Nastaví se sequence number na nulu. V případě zjištění datagramů připravených k odeslání provedou se tyto kroky:

1) Šablona AH je zařazena mezi IP a vyšší vrstvy.

2) Sequence numer se uloží do AH hlavičky. Ve stejném čase je zkontrolováno, jestli není v nepřípustném rozsahu. Pokud tomu tak je, AH vytvoří nový SA záznam a inicializuje hodnotu na nula. V každém případě se hodnota inkrementuje.

3) Dojde k vyplnění polí AH, které nejsou přípustné pro ICV.

4) Pokud je to zapotřebí, doplní se hlavičky tak, aby byly násobkem 32 bitů pro IPv4 (IPv6 násobkem 64 bitů).

5) Nepřípustná pole u IP hlavičky jsou naplněna nulami a je vypočítáno ICV.

6) ICV je uloženo do AH hlavičky a nepřípustná pole jsou vyplněny skutečnými hodnotami.

7) IP datagram je umístěn do fronty pro odeslání.

Příjem datagramu zvenčí se příliš neliší. Na začátku se čeká, zda byl fragmentován. Po složení celého datagramu na vstupu probíhá transformace z AH zabezpečení následujícím způsobem:

1) První na řadu přichází kontrola SPI a cílové IP adresy. Získané údaje se porovnají s SA uložených v SPD. Pokud není nalezena shoda, je datagram zahozen.

2) V dalším kroku je kontrola sequence number. Pokud se dané číslo liší od očekávané hodnoty, je datagram zahozen (antireplay filtr).

3) Nepřípustná pole v AH, tak i v IP hlavičce, se naplní nulami.

4) Autentizační algoritmus a klíč jsou použity pro výpočet ICV. Výsledek je porovnán s AH polem autentizační data. Pokud nedojde ke shodě paket, je zahozen.

5) AH hlavička je odstraněna a datagram již v původní podobě postupuje na výstupní frontu.

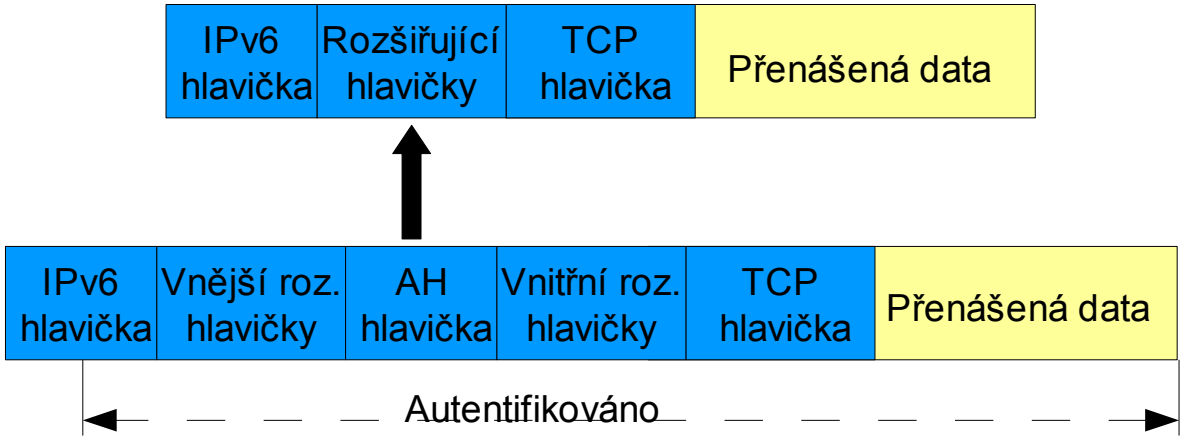
Stejně jako ESP, umožňuje AH provoz ve dvou módech a to tunelovacím a transportním. Princip funkce i vkládání hlaviček se v Ipv4, až na drobné detaily, neliší od zmiňovaného obecného principu.

4.6.1 AH na IPv6

Použití AH na protokolu IPv6 se od použití na IPv4 příliš neliší. První rozdíl spočívá v nepřípustných polích, která jsou jiné než u starší verze protokolu. Pole typu nepřípustná jsou zobrazeny na obrázku 14 žlutou barvou. Tyto pole jsou stejně jako u IPv4 před vypočítáním ICV inicializovány na nulu.

Obrázek 14: Nepřípustná pole pro výpočet AH v IPv6 hlavičce

Další rozdíl spočívá v umístění rozšiřující hlavičky. Z důvodu označení AH jako end-to-end protokol, následuje za hop-by-hop rozšířenou hlavičkou, místo, aby byla umístěna hned za IP hlavičku. Pokud je ale důvod umístit před nebo za AH jiné rozšiřující hlavičky, může být tak učiněno. To znamená, že zapouzdření s IPv6 je mírně komplikovanější. Celý proces zapouzdření v IPv6 transportním režimu je zobrazen na obrázku 15.



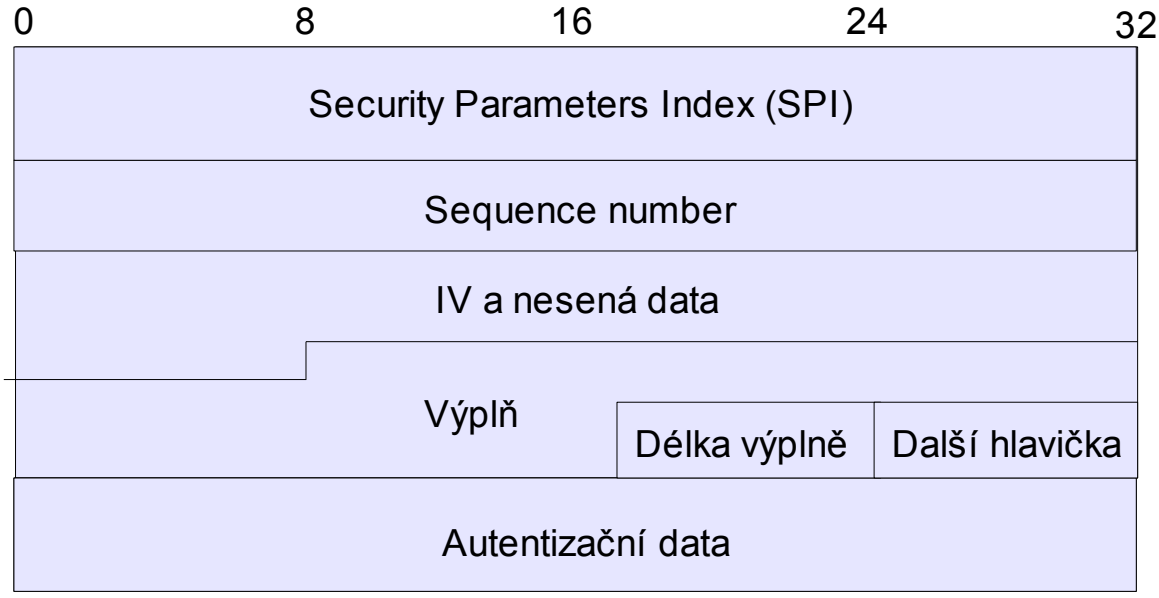
Obrázek 15: Použití AH v transportním režimu v IPv6

V tunelovém režimu je celý proces zapouzdření mnohem jednoduší, zde již nedochází k rozdělení rozšířených hlaviček na vnější a vnitřní, pouze se před AH, podle základního modelu, vytvoří hlavičky nové.

4.7 ESP (Encapsulating Security Payload)

Funkce ESP se využívá v případě potřeby provést u komunikace zabezpečení a to jak ověření komunikujících stran, autenticitu přenesených dat, tak i jejich šifrování. I přes to, že nám tato rozšířená hlavička umožňuje použít jak vlastnosti AH, tak šifrování, není problém využívat pouze jednu ze zmíněných možností. Způsob ověřování je, s výjimkou ověřovaných dat a s jejich umístěním, naprosto shodný jako v případě AH. Díky využití šifrování, je i velmi ztížené určit komu odchycený paket patří. V případě odchycení zašifrovaného paketu nelze určit, zda je transportován v tunelovém či transportním režimu. Díky této vlastnosti nejde například provádět nevyžádaný monitoring přenosu dat.

Na dalším obrázku 16 je zobrazena ESP hlavička spolu s Trailerem.



Obrázek 16: ESP hlavička s Trailerem.

Stejně jako u AH slouží SPI, cílová IP adresa a typ IPsec protokolu k jednoznačné identifikaci v pravidlech SA. Také zde se nachází sequence number. Ten stejně jako u předchozího varianty zabezpečení, slouží k ochraně proti replay útokům. Podobně se zachází i s inicializování této

hodnoty. Při vytvoření nové SA je nastavena na nulu. Při každém odeslaném datagramu je hodnota inkrementována o jedna. Pokud se hodnota dostane až ke $2^{32} - 1$, je ustanovena nová SA a vytvořeny nové autentizační klíče. Některé šifrovací algoritmy potřebují pro svoji funkčnost IV (inicializační vektor), zejména pokud se jedná o šifry pracující v CBC módu. Teoreticky by bylo nejlepší nechat poslat vektor pouze v prvním datagramu a nechat poslední blok v cache a tím rozšifrovat následující blok. V praxi a tedy v nespolehlivé síti, kdy není doručení zajištěno, se tato varianta jeví jako nepraktická a proto IPsec požaduje, aby byl vektor posílán v každém datagramu.

Často musí být ke správné funkčnosti blok šifer doplněn prostým textem na násobek velikosti bloku. Takové doplnění se umísťuje hned za nesená data v oblasti výplně. I v případě využití proudové šifry nebo jejího úplného vypuštění, je možné požadovat doplnění na násobek bloku z důvodu správné pozice pro další bloky v ESP hlavičce. Takováto výplň je realizována po sobě jdoucími inkrementovanými hodnotami. Následující pole označené jako Délka výplně uchovává hodnotu označující velikost výplně a může nabývat hodnot od 0 do 255. Pole Další hlavička zde definuje typ dat, jaké jsou obsaženy v IV a nesených datech. Kontrola integrity datagramu s výpočtem ICV je prováděna nad celým ESP datagramem, pouze autentizační pole je vynecháno. Nejčastěji jsou pro výpočet ICV použity šifry HMAC-MD5-96 a HMAC-SHA1-96. Každá metoda využívá pouze prvních 96 bitů z obou šifrovacích algoritmů. Omezení tohoto typu podá případnému útočníkovi ještě méně informací a tak se stává šifrování bezpečnějším.

Při příchodu datagramu do vstupní fronty, kde se řadí pro uplatnění SA politiky, je nutné rozhodnout, v jakém módu bude přenášen. Průběh takového procesu je popsán dále:

1) V SPD je hledána vhodná SA, která je shodná s použitými selektory.

2) Sequence number je zvýšeno o jedničku a umístěno do ESP hlavičky.

3) Vloží se výplň, pokud je to nezbytné a vyplní se pole Délka výplně a další hlavička. V případě použití šifry vyžadující IV, je vektor přidán na konec nesených dat. Poté vektor, nesené data a ESP Trailer jsou zašifrovány příslušným algoritmem a klíčem uvedeným v SA.

4) Za pomocí klíče a použitého algoritmu v SA se vypočítá ICV a vloží se do pole Autentizační data.

5) Pokud výsledný datagram vyžaduje fragmentovat, musí být provedena v tomto bodě. V transportním módu je ESP aplikováno pouze na celý datagram. V tunelovém módu může být použito ESP na fragmenty datagramu.

Pořadí, ve kterém jsou jednotlivé kroky prováděny, je důležité a to z hlediska značné náročnosti dešifrování dat oproti ověření pravosti přijatých dat. V případě, že by ověření selhalo, bude paket zahozen bez užitku a tak by čas strávený procesorem na dešifrování byl promarněn. V situaci, kdy dorazí datagram do vstupní fronty v zašifrované podobě, je s ním naloženo následujícím postupem:

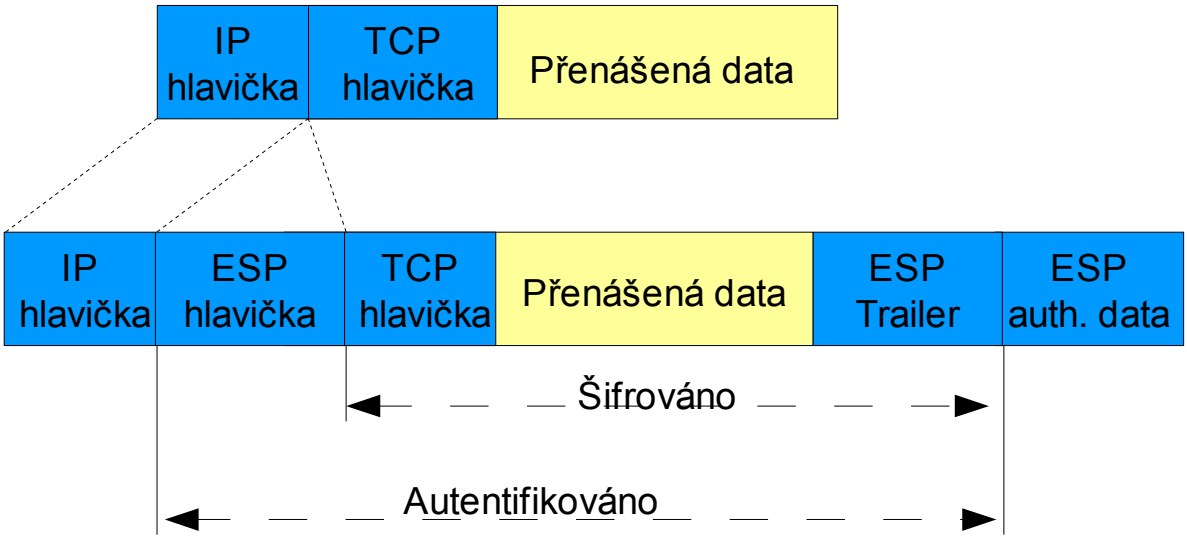
1) Zkontroluje se shoda SPI a cílové IP adresy s SA. Pokud není nalezena, datagram je zahozen.

2) V případě kdy je aktivována ochrana proti replay útokům, provádí se kontrola sequence number, je hodnota ověřena a s datagramem je podle toho naloženo.

3) Dalším krokem je vypočítat ICV z ESP hlavičky, nesených dat a pole ESP Trailer, za pomoci klíče a algoritmu uvedených v SA. V případě neshody, je datagram zahozen a je aktualizováno antireplay okno.

4) Pole nesených dat a ESP Trailer jsou dešifrovány pomocí algoritmu a klíče uvedených v SA. Pokud byla přidána výplň, je zkontrolována, zda je vhodná pro dešifrovací algoritmus. Datagram je zrekonstruován a předán na výstupní frontu k dalšímu zpracování, které je závislé na tom, zda byl použit transportní či tunelovací režim.

V transportním módu se ESP používá pro zajištění bezpečnosti vyšších vrstev IP datagramů. Nejčastěji se jedná o segment TCP, UDP nebo ICMP datagram. Na obrázku 17 je zobrazeno ESP zapouzdření pro TCP segment. V tomto případě se ESP hlavička vkládá hned za IP hlavičku, ale ještě před TCP hlavičku.



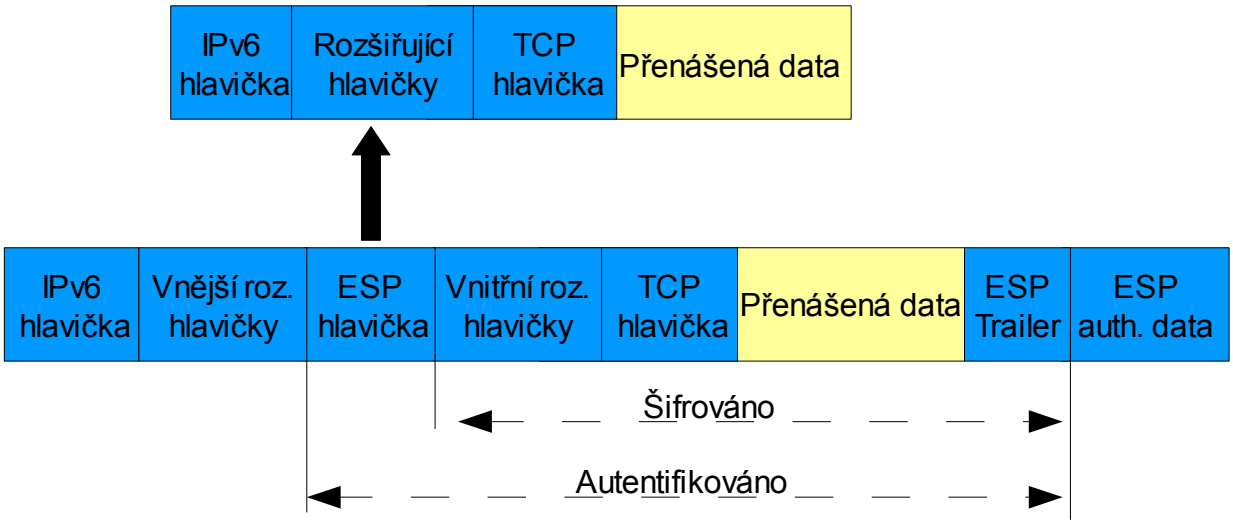
Obrázek 17: Zapouzdření pomocí ESP v transportním režimu

V tunelovacím režimu se ESP chová jako v modelovém případě, tedy dojde k zapouzdření celého datagramu i s TCP a IP hlavičkami. Po zapouzdření a tedy i zašifrování původní IP hlavičky, TCP hlavičky, přenášených dat a ESP Traileru se vytvoří nová IP hlavička, která již nese údaje brány, na které bylo IPsec zabezpečení provedeno a cílové brány kde dojde k rozšifrování.

4.7.1 ESP na IPv6

Provoz ESP v IPv6 je velmi totožný s provozem v Ipv4, jediný rozdíl spočívá v umístění rozšířených hlaviček. Podobně jako při použití AH, se liší jejich umístění podle toho, v kterém módu jsou data přenášena. V případě zapouzdření pomocí ESP, musí být hop-by-hop volby a informace o routingu přístupné i během přenosu a tak by se měly nacházet v nezašifrované části datagramu. V praxi to znamená, že některé rozšířené hlavičky přijdou před nebo po ESP hlavičce. Je dobrým zvykem všechny hlavičky, které nejsou nutné pro přenos, umístit do šifrované části datagramu, tedy za ESP hlavičku.

Obrázek 18 nám ukazuje jak je datagram zapouzdřen v transportním režimu. Rozšířené hlavičky jsou umístěné do dvou míst a to před a po ESP hlavičce. Pole další hlavička v ESP Traileru bude mít typ rozšířené hlavičky následující ihned za ESP hlavičkou.



Obrázek 18: Datagram zapouzdřený pomocí ESP v IPv6

Při použití tunelovacího režimu ESP se celý proces zabaleni a umístění hlaviček nijak neliší od použití v IPv4.

4.8 Internet Key Exchange (IKE)

Mezi nejdůležitější procesy při požadavku o šifrování či ověření pravosti přijatých dat, je způsob výměny a následná správa klíčů. V takovém případě můžeme zvolit ze dvou přístupů. Jedním z nich je manuální konfigurace. Ta má výhodu v počáteční rychlosti konfigurace a celkové transparentnosti celého procesu, ale při větším počtu spojení se náročnost konfigurace značně zvyšuje a stává se neudržitelnou. Další nevýhodu, kterou přináší manuální konfigurace, je nemožnost měnit v dostatečně krátkém čase šifrovací klíče a tak i při prolomení šifry útočníkovi zpřístupnit jen malou část komunikace. Druhým přístupem je dynamická konfigurace. V dřívější době jako stěžejní protokoly byly používané ISAKMP a IKEv1. Kde ISAKMP (Internet Security Association and Key Management Protocol) se staral o dohodu o použitých šifrovacích algoritmech a nastavení dalších parametrů SA. Druhý protokol IKE (Internet Key Exchange) ve verzi 1 měl na starosti výměnu klíčů pro šifrovací algoritmy. I přes to, že daný přístup byl funkční, nejevil se zrovna jako nejlepší a proto se jak správa SA, tak výměna bezpečnostních klíčů, zastřešila pod protokol IKE ve verzi 2.

Základní myšlenka IKE je velmi jednoduchá, pomocí Diffie-Hellman algoritmu pro bezpečnou výměnu klíčů tuto výměnu provézt a dále pomocí nich komunikaci zabezpečit. Bohužel, i zde jsou překážky, které musí IKE zabezpečit. Mezi ně můžeme zařadit replay útoky, man-in-the-middle a další, které se pokouší nabourat bezpečnou výměnu klíčů. IKE je často označován jako hybridní protokol, protože jeho základní vlastnosti jsou odvozeny od dalších třech protokolů.

První protokol, použitý jako předloha, se nazývá ISAKMP. Ten se používal v prvních implementacích IPsec a zde odvozené vlastnosti jsou použití rámce, který stanoví mechanizmus a formát zpráv pro další používané pro-

tokoly. Ty mají poté za úkol vytvořit SA a provedou výměnu klíčů. Velkou výhodou ISAKMP je nezávislost na konkrétních metodách pro výměnu klíčů a funkčností pouze obecného rámce může obsahovat téměř jakýkoliv algoritmus. Pro výměnu klíčů používá IKE metody, které jsou založeny na Diffie-Hellman algoritmu. Metody tohoto typu se řadí do skupiny, která se nazývá OAKLEY. Dále tyto metody zajišťují ověřování identity, autentizaci a zpětné předání zabezpečujících informací. V IKE jsou odvozeny i vlastnosti z protokolu SKEME. Je to protokol na univerzální výměnu klíčů, který poskytuje výbornou bezpečnost přenášeného tajemství, výměnu klíčů a dohodu o používaných kryptografických algoritmech. IKE si vzala od jmenovaného protokolu metodu užití veřejného klíče pro šifrování a autentizaci a myšlenku rychlé změny klíče při přenosu.

4.8.1 ISAKMP

Znalost principu funkce tohoto protokolu je velmi důležitá i přesto, že se od něho v poslední implementaci opustilo. Z důvod poskytování IKE základní mechanismy a formáty zpráv. ISAKMP vytváří SA záznamy ve dvou fázích. V první fázi se ověří komunikující strany a vytvoří se bezpečné spojení mezi nimi. Ve druhé fázi vyjedná SA pro danou VPN. Při použití IKE je se například použití ESP a AH sjednává až ve druhé fázi a to za pomocí zabezpečeného kanálu z fáze první.

Jedním z cílů ISAKMP je poskytnout ochranu proti takzvaným DOS (denial-of-service) útokům. Bohužel, ani zde, se není možné absolutně ubránit. Jeden z útoků, který se může stát za určitých okolností nepříjemný, je zaslání velkého množství podvržených paketů obsahujících IKE informace s náhodně generovanými IP adresami. To má za následek přetížení přístupového zařízení. To je způsobeno díky použití Diffie-Hellman algoritmu, který je značně výpočetně náročný. V případě přetížení je zabráněno zpracování

legitimních informací a tak navázání nového tunelového spojení. Potlačení efektu takových útokům je možné pomocí ISAKMP cookies. Ty jsou ještě před započetím náročných výpočtů algoritmů zaslány na adresu, s níž chtějí komunikovat a tak uzel ví, že příchozí požadavek, který má stejné cookies je legitimní a může začít s obsluhou příchozích požadavků. Zde popisované cookies musí splňovat mnoho podmínek, aby se skutečně jednalo o prostředek, který má daný problém řešit.

– Musí být vázány jen na konkrétní dva uživatele, jenž hodlají spolu komunikovat. To znamená, že musí být unikátní, aby nebylo možné je získat například z jiného připojení.

– Nelze je vygenerovat žádný běžným způsobem bez specifických informací známých jen účastníkům komunikace.

– Doba použití je omezena pouze na jedinou komunikaci.

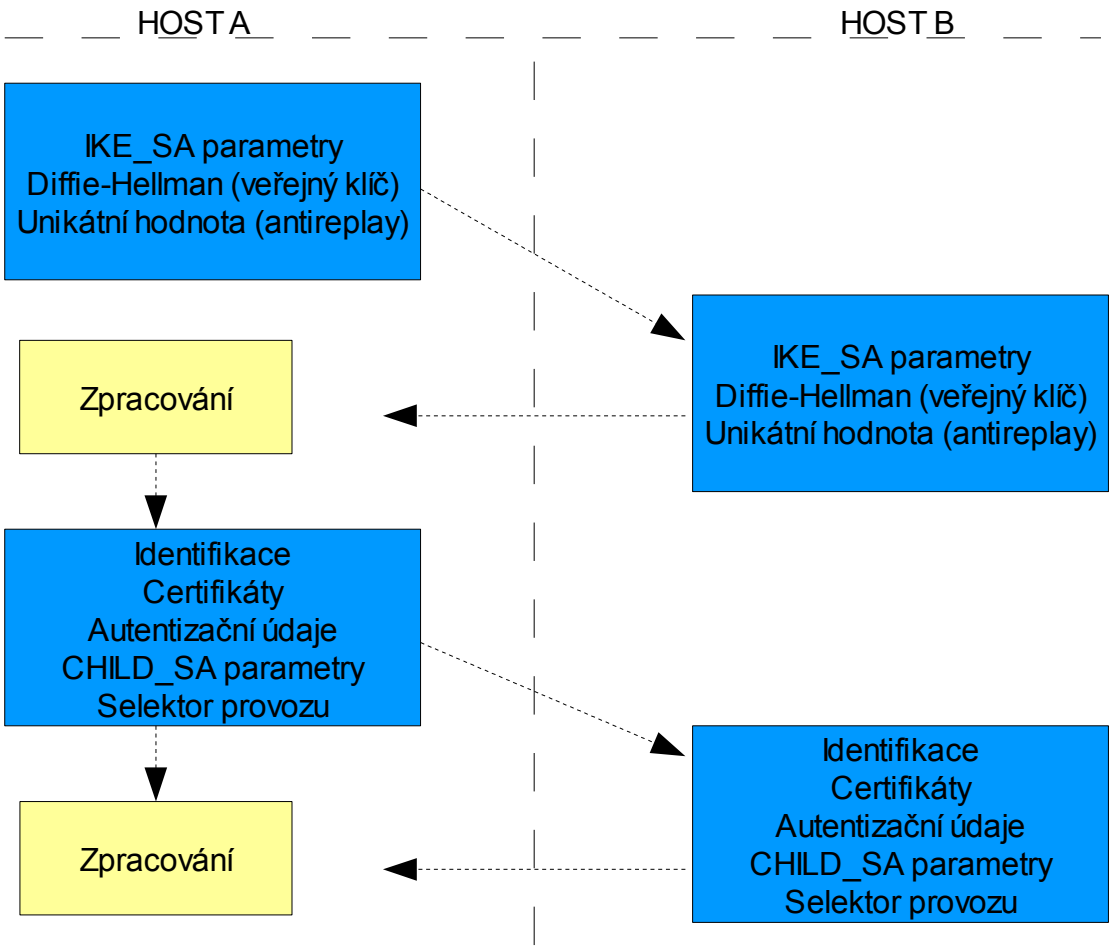
4.8.2 IKEv2 (Internet Key Exchange version 2)

V případě poslední specifikace IPsec probíhají výměny klíčů za použití IKEv2. Funguje na principu odeslání požadavku a čekání na odpověď. Pokud odpověď do určité doby nedorazí, požadavek se jednoduše zopakuje. Pro komunikaci se používá nezaručený protokol UDP s porty 500 a 4500. IKEv2 je popsáno v RFC 4306, kde jsou definovány čtyři fáze výměny klíčů. První dvě fáze sloužící pro zahájení nazýváme IKE_SA (IKE_SA_INIT) a IKE_AUTH. Další fáze slouží ke správě SA a je označena CREATE_CHILD_SA a poslední slouží k výměně informací a nazývá se INFORMATIONAL. První dvě fáze mohou proběhnout vždy jen jednou a další mohou začít vždy až po jejich skončení. CREATE_CHILD_SA může být libovolný počet a ani nezáleží na jejich pořadí.

První žádost ve fázi IKE_SA_INIT vyjednává bezpečnostní mechanizmus, použitý protokol a Diffie-Hellman hodnoty. Zpráva tedy obsahuje návrh bezpečnostního mechanizmu pro SA a používaný protokol. Zde můžeme volit mezi AH, ESP nebo oběma. Dále se definuje kryptografický algoritmus, který může být doplněný o atributy k použitému algoritmu. Po splnění těchto úkonů se vytvoří první IKE_SA. Položkou, která musí být ještě obsažena v IKE_SA_INIT je výměna klíčů pomocí Diffie-Hellman algoritmu.

Tento algoritmus pracuje na velmi důmyslném principu. Nejprve si každý z účastníků vygeneruje náhodné číslo a uloží si ho jako tajnou hodnotu. Z této hodnoty vypočítá druhou hodnotu (veřejný klíč) a pošle ji protější straně s níž chce navázat zabezpečenou komunikaci. Z druhé strany přijde taktéž veřejný klíč. A díky jednosměrnému matematickému výpočtu, založenému na kombinaci svého tajného klíče s veřejným klíčem protějšku, dostanou stejný výsledek. Ten však již nelze odvodit z veřejných klíčů přenášených mezi sebou a tak obě strany mají stejné číslo, podle něhož se odvodí klíč pro šifrovací algoritmy.

Celý proces komunikace v první a druhé fázi je zobrazen na obrázku 19. Je zde zachyceno odeslání první zprávy spolu se všemi přenášenými údaji a odpověď na tuto zprávu. Dále je zobrazeno odeslání třetí zprávy směrem od iniciátora spojení. Tato zpráva má za úkol provést druhou fázi IKE_AUTH, to znamená ověřit uživatele a vytvořit první datovou SA.

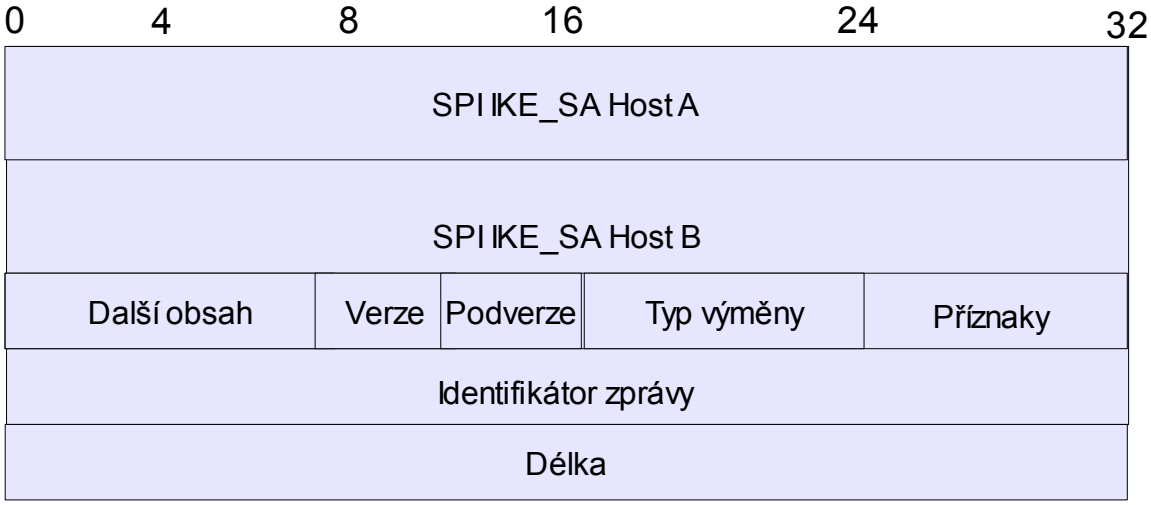


Obrázek 19: První dvě fáze navazování spojení IKEv2

Ještě před vytvořením datové SA, značené jako CHILD_SA, je nutné provést identifikaci účastníků. Označení pro tuto fází je IKE_AUTH, v ní již jsou poslána první autentizační data a informace o vlastní identitě, tím prokáže znalost tajného klíče pro danou identitu. Tento požadavek může obsahovat i sadu certifikátů pro usnadnění ověření účastníka. Další údaje obsažené v komunikaci, jsou parametry prvního potomka IKE_SA a díky takzvaným selektorům může přidat pravidla pro rozsah adres, které mohou být použity v komunikaci nebo pakety na něž se mají dané pravidla vztahovat. Protějšek analogicky na dotaz odpoví a komunikace může být po pouhých 4 zprávách navázána. Což je obrovský pokrok vůči dřívějšímu řešení.

Kdykoliv později, když je potřeba změnit šifrovací klíče nebo vytvořit novou SA, použije se pouze fáze CREATE_CHILD_SA. Tuto fázi může zahájit kdokoliv z komunikujících stran.

V případě zpráv, generovaných IKEv2, to není příliš jednoznačné. Všechny zprávy mají pouze definované bloky, jenž může zpráva obsahovat, ale jejich pořadí je nevýznamné. Ve většině případů se jedná o stejné složení, založené na typu zprávy, kterou chci odeslat. Jediným společným identifikátorem, který ukazuje na zprávu IKEv2, je hlavička IKE. Ta je zobrazena na obrázku 20.



Obrázek 20: Hlavička IKEv2

Pole typ výměny označuje kódem o jaký typ komunikace se jedná na výběr zde máme IKE_SA_INIT, IKE_AUTH, CREATE_CHIL_SA, INFORMATIONAL a rozsah hodnot rezervován pro budoucí a privátní využití. Identifikátor zprávy zde slouží pro označení odeslané zprávy a současné přiřazení přijaté zprávy k odeslané. Poslední zajímavé pole je nazývané příznaky. V tomto poli je specifikováno zda je zpráva požadavek či odpověď, zda podporuje vyšší verzi a zprávu posílá Host A nebo Host B.

Poslední problém, na který můžeme narazit ve spojení s navazováním komunikace, je autentizace. V případě manuální konfigurace, pokud můžeme zaručit, že klíč nebyl ukraden či zkopírován ze zařízení, kde je použit, není důvod se o autentizaci starat. Bohužel, v případě využití dynamických prostředků je problém vážnější. Cestou mohou být takzvané certifikáty. Ty jsou vydávány certifikačními autoritami (CA). Celý princip spočívá v tom, že musí obě komunikující straně CA důvěřovat. Při splnění požadavku důvěry je u strany navazující komunikaci přidán k veřejnému klíči certifikát, který je opatřen digitálním podpisem vycházejícím ze soukromého klíče. Bohužel těchto certifikačních autorit je velké množství a jen málokdy jsou dva certifikáty od různých CA kompatibilní. To má za následek nemožnost takového zabezpečení využít ve větším měřítku se 100 procentní jistotou. Jako řešení by se mohlo jevit EAP. Jedná se podobně jako u ISAKMP pouze o obal, který definuje formát zpráv a obaluje další autentizační protokoly. V současné době je jeho využití například v bezdrátových sítích, kde se používá k ověření připojujících se uživatelů.

Na závěr uvádím tabulku 3 s odkazem na jednotlivá RFC (request for comments), v nichž je uvedený detailní popis každé části věnující se IPsec a IPv6.

Tabulka 3: RFC vztahující se k problematice IPsec a IPv6. Zdroj: Vlastní

RFC 2401	Security Architecture for the Internet Protocol
RFC 2402	IP Authentication Header
RFC 2404	The Use of Hash Message Authentication Code Federal Information Processing Standard 180-1 within Encapsulating Security Payload and Authentication Header
RFC 2406	IP Encapsulating Security Payload (ESP)
RFC 2407	The Internet Security Domain of Interpretation for ISAKMP
RFC 2408	Internet Security Association and Key Management Protocol (ISAKMP)
RFC 2409	Internet Key Exchange (IKE)
RFC 2460	Internet Protocol, Version 6 (IPv6) Specification
RFC 2474	Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers
RFC 3576	Change of Authorization
RFC 4109	Algorithms for Internet Key Exchange version 1 (IKEv1)
RFC 4301	Security Architecture for the Internet Protocol
RFC 4302	IP Authentication Header
RFC 4303	IP Encapsulating Security Payload (ESP)
RFC 4306	Internet Key Exchange (IKEv2) Protocol
RFC 4308	Cryptographic Suites for IPsec

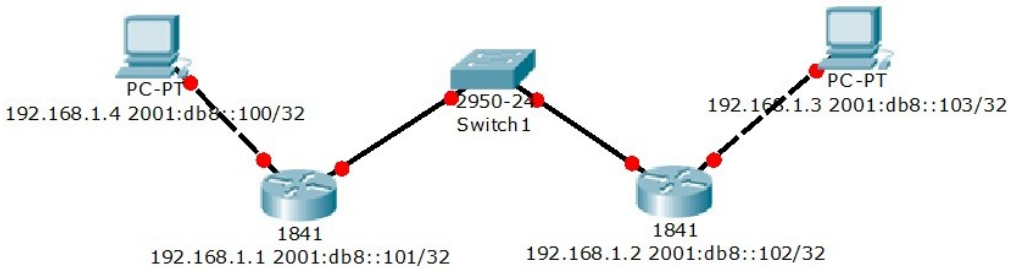
5 Praktická část

5.1 Testovací konfigurace

Testovací konfigurace pro transportní režim je zobrazena na obrázku 21. Nalézají se zde dva počítače spojeny běžnou internetovou linkou kde switch má zobrazovat neidentifikované prostředí mezi nimi.

Obrázek 21: Testovací konfigurace transportního režimu

Na dalším obrázku 22 je zobrazena virtuální konfigurace použitá pří testování IPsec v tunelovém módu.



Obrázek 22: Testovací konfigurace pro Tunelový mód

V případě provádění testů je daný nákres pro tunelový mód pouze ukázkový. Při testech obstarával roli gateway, která má šifrovat přenos ten samí počítač na který byli pakety směrovány. Důvod k takovému opatření byl testování všech konfigurací na virtuálních strojích. Bohužel emulace 4 zároveň běžících strojů vyžaduje opravdu výkonný hardware.

5.1.1 IPsec na Windows

Použití IPsec na systému od společnosti Microsoft je bezproblémovější než by se mohlo zdát (tedy alespoň v IPv4). První implementace se nacházela v systému Windows 2000, kde musela být doinstalována. Bohužel v tom samém systému je podpora IPv6 téměř nepoužitelná a musí se složitě doinstalovávat. První systém s použitelnou podporou je Windows XP. I zde se jedná spíše o vývojovou verzi Microsoftu než o plnohodnotného klienta. Microsoft vydal i aktualizaci označovanou zlepšení funkčnosti a rozšíření možností IPsec. Funkčnost této „záplaty“ je nesporná. Bez ní je navázání bezpečného spojení jen velmi obtížné a je značně nestabilní. Bohužel rozšíření ve směru podporovaných algoritmů se nekoná.

V případě konfigurace spojení máte na výběr u Windows XP z obou módů. Tedy jak tunelového tak transportního. V praxi je jediný využitelný mód tunelující. Další omezení, které nám klade implementace systému je podpora AH a ESP. V případě, že chcete využít IPv6 IPsec máte možnost pouze autentizace za použití md5. Šifrování v tomto režimu není podporováno. IKE spolu s IPv6 pozbývá funkčnosti. Další nepříjemné zjištění se koná v případě nastavení politiky přenosu. Nastavení pro IPv6 nelze provádět pomocí Windows XP IPSec Policies snap–in, ale musí se použít netradičně (u Windows) konzolová aplikace Ipsec6.exe command–line tool. Bohu-

žel ani nastavení za její pomocí se nejeví jako použitelný prostředek a proto při použití IPv6 na Windows XP je lepší zvolit externí řešení.

V Případě IPv4 je situace jiná. Zde je IPsec v systému použitelný a poměrně snadno konfigurovatelný. Na výběr máme jak šifrování pomocí ESP a algoritmů DES a 3dDES tak AH s algoritmy SHA1 a MD5. Všechny ze zmiňovaných algoritmů jsou funkční a nic nebrání jejich použití. IKE je v případě IPv4 také použitelné i když pouze ve staré verzi využívající jako řídící protokol ISAKMP. Máme zde na výběr ze třech možností, buď zvolíme Kerberos, Certifikáty a nebo předsdílený klíč.

Systém Windows Vista prošel řádnou modernizací od předchozího systému. Již nabízí plnou podporu pro IPv6. Ta je aktivní ihned při prvním spuštění systému. Připojení do IPv6 je provedeno skrz klienta od Microsoftu nazývaného Teredo. Klient se při konektivitě do internetu připojí na tunelovací server výrobce a ihned nabídne IPv6 spojení do okolní sítě. V případě použití IPsec nejspíše nenarazíte na žádný problém. Implementace je zde již vydařená a nově nabízí podporu algoritmů ECDH (Elliptic Curve Diffie-Hellman) pro autentizaci a AES pro šifrování. Oba tyto algoritmy zaručují vysoký standart při zabezpečení. Jejich využití je dnes velmi omezené jen málo implementací je používá. Z dílen Microsoftu máme možnost použití pouze ve spojení s Windows server 2008. Nově také přibilo nové rozhraní pro okenní konfiguraci nazývané Brána firewall s vyspělým zabezpečením. S jeho pomocí je konfigurace IPsec zabezpečeného spojení opravdu otázkou pár kliknutí. Způsob jak tímto nástrojem vytvořit politiky pro zabezpečený kanál je popsán v příloze číslo 2 C. Samozřejmě lze využít i starší nástroj IPSec Policies snap–in, který nově nabízí podporu pro IPv6. Bohužel se občas stávalo, že při použití jednoho či druhého principu vytváření IPsec politik nedošlo k jejich aktivaci. V takovém případě se musel vytáhnout síťový kabel

(nejrychlejší řešení) a nebo restartovat počítač. Myslím, že zde šlo pouze o nějaký bug, který bude snadno opravitelný záplatou.

Poslední testovaný systém od Microsoftu byl Windows server 2003. Při testování byla použita verze bez prvního service packu pouze s nabízenými updaty pro IPsec. Jedná o stejný druh implementace jako v případě Windows XP. Při použití protokolu IPv6 nemáte možnost využití jiné šifry pro ochranu klíče než MD5 s protokolem IPsec.

5.1.2 IPsec na Linuxu

V Linuxu je podpora IPv6 již od roku 1998 spolu s jádrem 2.1.8. Plná verze má označení 2.6.12 a obsahuje již stabilní implementaci IPv6. Díky tak dlouhé podpoře je široká podpora pro tento protokol, spolu s velkým množstvím nabízených aplikací tedy alespoň ve srovnání s platformou Windows OS. Podpora IPsec je také na dobré úrovni. Když jádro vašeho systému nese označení 2.5 a novější je již podpora implementována. K využití a vytvoření spojení nyní stačí již jen ipsec-tools vyvíjeny společností Novell. V dnešní době podporují ipsec-tools téměř všechny používané algoritmy jak pro autentizaci tak pro šifrování. Samozřejmostí je podpora těchto algoritmů jak v IPv4 tak v IPv6. Není problém zprovoznit tunelový i transportního mód. Způsob zprovoznění tunelového i transportního přenosu je popsán v příloze číslo 2 A. Podpora IKE je zajištěna díky portovanému démonu Racoon. Ten slouží pro automatickou výměnu klíčů a nastavení parametrů spojení. Racoon podporuje ověřování (authentication) pomocí sdílených klíčů, X.509 certifikatů a Kerberosu. Díky tak široké paletě nabízených funkcí téměř nenastane chvíle kdy je ipsec-tools v koncích. Bohužel to otevírá prostor pro situace kdy není protější strana schopna akceptovat vaše nastavení. V takovém případě je nutné zahájit celý proces vyjednávání o použitých algoritmech znovu s použitím „slabších“ algoritmů. To se zejména projevuje při spojení s

OS Windows. V posledních generacích Windows je implementace solidní, ale podpora nových šifrovacích algoritmů je velmi pozvolná. Za celou dobu testování se nestalo, že by hendikep byla implementace na systému Linux. Pouze démon racoon nechtěl pracovat střídavě na jednom či druhém systému v propojení Linux-Linux. Tato chyba se nepodařila vyřešit za dobu testování. Nejspíše byl problém v některém z nastavení operačního sytému což vedlo k těmto chybám. Důvodem mohlo být i klonování operačních systémů v prostředí WMware, které činilo zejména problémy při vytváření klonů systému Windows.

5.1.3 Použití komerčních variant s IP-sec

Podpora IPsec v jádru systému se dá také obejít za pomocí klienta, který spojení vytvoří, udržuje a řídí sám. Takovým příkladem je Cisco VPN client. Jde o produkt, který je volně ke stáhnutí a nabízí nám zabezpečené spojení pomocí IPsec a to jak v tunelovém tak transportním režimu. Je portován téměř na všechny systémy, takže jeho využití nic nebrání. Dále umožňuje i využití IPv6 sítí, ale zde je závislí na podpoře v systému. Pro stranu serveru je implicitně určený produkt od Cisca s názvem Cisco VPN koncentrátor. Ten má za úkol obsluhovat příchozí IPsec spojení a rozhodnout podle konfigurace jak má být s nimi naloženo.

Dále uvádím tabulky 4 a 5 pro možnosti srovnání jednotlivých řešení ze strany operačních systémů.

Tabulka 4: Podporované algoritmy v operačních systémech. Zdroj: Vlastní

Operační systém	Podporované algoritmy v IPv4		Podporované algoritmy v IPv6	
	Autentizace	Šifrování	Autentizace	Šifrování
Windows XP Windows 2003 server	hmac-sha1, hmac-md5	des-cbc, 3des-cbc	hmac- MD5	nepodporováno
Windows Vista	hmac-sha1, hmac-md5, hmac-ecdh	des-cbc, 3des-cbc, aes-ctr	hmac-sha1, hmac-md5, hmac-ecdh	des-cbc, 3des-cbc, aes-ctr
Linux Ubuntu 9.04	hmac-sha1, hmac- md5, keyed-md5, keyed-sha1, hmac-sha(až 512), hmac-ripemd160, aes-xcbc-mac, tcp-md5	des-cbc, 3des-cbc, blowfish-cbc, cast128-cbc, des-deriv, rijndael-cbc, twofish-cbc, aes-ctr	hmac-sha1, hmac- md5, keyed-md5, keyed-sha1, hmac-sha(až 512), hmac-ripemd160, aes-xcbc-mac, tcp-md5	des-cbc, 3des-cbc, blowfish-cbc, cast128-cbc, des-deriv, rijndael-cbc, twofish-cbc, aes-ctr
Cisco VPN client	hmac-sha1, hmac-md5	des-cbc, 3des-cbc, aes-ctr	hmac-sha1, hmac-md5	des-cbc, 3des-cbc, aes-ctr

Tabulka 5: Podpora IPsec a IKE v závislosti na použitém protokolu. Zdroj: Vlastní

Operační systém	Podpora IKE	IPsec v IPv4	IPsec v IPv6
Windows XP Windows 2003 server	Ano pouze v IPv4	Ano	ANO pouze autentizace
Windows Vista	Ano	Ano	Ano
Linux Ubuntu 9.04	Ano	Ano	Ano
Cisco VPN client	Ano	Ano	Ano

6 Možnosti připojení do sítě využívající IPv6

V dnešní době je již téměř ve všech operačních systémech a síťovém hardware dobrá podpora pro IPv6. Tím máme spoustu možností pro připojení do nového typu sítě. Mezi základní typy a určitě postačující řešení i pro středně dlouhý testovací provoz, je připojit se skrz klienta umístěného v operačním systému nebo přes jeden z otevřených tunelových serverů. V případě stabilního dlouhodobém provozu má již smysl uvažovat o jedné z možnosti trvalého připojení. Mezi základní způsob a zároveň i nejvhodnější řešení ve všech směrech je podpora ze strany poskytovatele. Další možnosti jsou získat po registraci na Freenet6 nebo SixXS vlastní rozsah IPv6 adres a vytvořit si tak trvalé připojení. Poslední použitelnou variantou je 6to4 automatické tunelování.

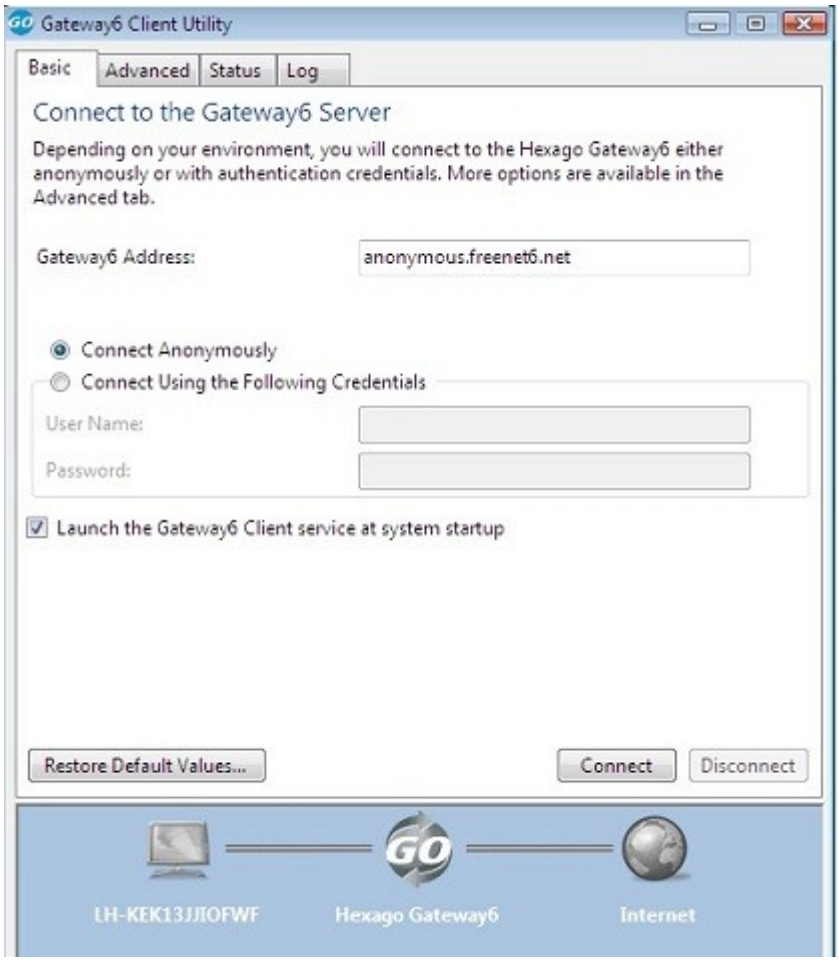
Základní typy připojení do IPv6 jsou velmi dobře použitelné v raných fázích testování provozu v nové síti. I když v dnešní době, zejména v případě klienta od freenet6, se jedná o solidní řešení. Oba způsoby, jak klient od freenet6, tak klient v obsažený v systému, se dobře vyrovnávají s překážkami současné sítě IPv4 jako jsou NAT a firewall.

V případě užívání systému Windows je možné použít klienta nazývaného Teredo. Ten je již automatickou součástí systému Windows Vista a do systému Windows XP ho není problém doinstalovat. Po spuštění klienta Teredo dojde k vytvoření konexe na Teredo server společnosti Microsoft a dojde k zpřístupnění IPv6 sítě skrze jejich server. Bohužel podle zkušeností se nejedná o příliš dobrý způsob. Spojení se často nenaváže a nebo se tváří jako nefunkční, ale přitom má přidělenou IP adresu od serveru. Negativní zkušenosti jsou zejména při použití klienta na systému Windows XP, který byl spíše průkopníkem v oblasti IPv6 než reálně použivatelnou verzí. V systému Windows Vista je provoz Tereda téměř bezproblémový. Sice také trpí některými „dětskými“ nemocemi jako v případě win xp. I přes to se jedná o použitelné řešení, alespoň pro začátek. Překvapivě je Teredo implementováno v serverové edici Windows a to konkrétně Windows server 2003 a 2008. Použití ve verzi 2003 se shoduje s implementací ve Windows Vista.

Klient Teredo je portován i na ostatní systémy a to jak Linux, tak na MacOS X pod názvem Miredo. Zvláštností je, že i když se jedná o ekvivalent Tereda, funkčnost na Linuxu se zdála mnohem lepší než v případě systému Windows. Zejména byl vidět rozdíl v rychlosti odezvy a celkové stability provozu.

Další ze základních možností připojení je skrz klienta Freenet6. Tento tunelovací server umožňuje i práci v anonymním režimu, což téměř nuluje potřebu konfigurace. Klient pro připojení je portován téměř na všechny sys-

témy a to jak typu Windows tak Unix. Na obrázku 23 je zobrazen klient Freenet6 určený pro Windows.



Obrázek 23: Klient pro připojení do IPv6 od Freenet6

Připojení k síti je velmi rychlé a i přes to, že pakety neputují zrovna ideální cestou, se jedná o vcelku rychlé a stabilní připojení. V případě porovnání s klientem od Microsoftu se jedná rozhodně o lepší variantu připojení do IPv6.

Nevýhoda obou základních typů připojení je očekávání, že se o veškerou komunikaci budete pokoušet sami a tak připojení na vás z okolního internetu není téměř možné, pokud už nastane, daň za něj je téměř nulová spolehlivost a stabilita.

Trvalé formy připojení do IPv6 jsou již zajímavé z pohledu dlouhodobého užívání. Většinou poskytují výbornou rychlost, svižnou odezvu a stabilitu blížící se stabilitě vaší sítě.

V případě získání IPv6 adresy od vašeho poskytovatele se skutečně jedná o parametry vaší sítě, které tvoří ve všech směrech omezující faktor. Je nepříjemné, že se jedná v dnešní době spíše o vzácnost podpory IPv6 ze strany vašeho lokálního poskytovatele. A v případě, kdy skutečně podporuje nový protokol i váš poskytovatel, bývá kamenem úrazu některé ze zařízení na cestě od jeho směrovače k vám. Ze stránek ww.nix.cz lze zjistit, zda váš poskytovatel protokol podporuje. Pokud se tak stalo, můžete se plnohodnotně připojit do sítě. To znamená přenášení IPv6 datagramů a využívat IPv6 adresy z přiděleného rozsahu. V případě využití IPv6 adresy by neměl vzniknout žádný problém na jakémkoliv dnes používaném systému.

Velmi pravděpodobné je u zmiňované varianty náraz na překážku, jenž nelze obejít a zde přichází na řadu tunelování k tunelovacímu serveru. Tunnel Brokers, kteří umožňují trvalé připojení, je celá řada. Mezi nejzajímavější z hlediska využití IPv6 v Čechách patří Freenet6 a SixXS, kteří mají umístěné servery v Evropě a tak se dá dosáhnout dobré rychlosti i odezvy.

První z nich, Freenet6, nabízí jak anonymní režim, tak po registraci je možné zažádat o poskytnutí pevného rozsahu IP adres. Takto získaná adresa se nemění ani při změně IPv4 adresy. Získáte IPv6 adresu, kde je prefix v délce 48 bitů. Tím máte možnost užít IPv6 i ve své LAN a zničit tak překážky typu NAT. Poslední související vlastností je registrace vašeho tunelu do DNS a tím umožnění adresování všech počítačů ve vaší LAN zvenčí.

Další zmiňovaný tunnel brokers je SixXS. V jeho případě je registrace, co se týče získaných hodnot podobná jako v případě Freenet6. Registrace do jeho sítě je velmi důsledná, například při registraci nelze zadat žádná z free-

mailových adres a jejich databáze je opravdu rozsáhlá. Dále se vyplňuje důvod, proč se chcete připojit k IPv6 a mnoho dalšího. Výhoda zde spočívá například v možnosti si při registraci vybrat, na který server se budete chtít připojit a tak dosáhnout optimálního přístupu. Podobně i jako Freenet6 používá SixXS svého klienta pro připojení. Klient se nazývá Aiccu a pro komunikaci se servem používá svůj vlastní tunelovací protokol TIC (Tunnel Information and Control protocol). Nabídka portací na různé OS se zdá být úplná a tak by neměl nastat sebemenší problém při připojení do jejich sítě.

S přístupem tohoto typu, kdy tunelujete ze své přístupové stanice IPv6 skrz běžnou síť na vzdálený tunelovací server, by neměl nastat žádný problém u všech dnes používaných systémů.

Varianta připojení skrz 6to4 přichází v úvahu pouze ve chvíli, kdy máte veřejnou IPv4 adresu. Automatické tunelování 6to4 je založeno na principu vytvoření tunelového spojení, oproti veřejnému serveru, který disponuje nativní konektivitou do IPv6. Adresa je získána z veřejné IP adresy a to přidáním před 32 bitovou IPv4 adresu samé nuly tak, abychom dostali 128 bitovou délku. Základem pro 6to4 je přidělení prefixu z rozsahu 2002::/16. Z tohoto rozsahu bude poté tvořena adresa naší sítě a to nejprve prefix 2002, poté bude následovat naše IPv4 adresa v hexadecimální podobě. Podpora ze strany OS je podobně, jako u ostatních trvalých způsobů připojení, bezproblémová a to i z pohledu IPv6, prehistorickém systému Windows XP.

I když to na první pohled nemusí být zřejmé, použití IPv6 přinese sebou další náklady spojené s bezpečností jednotlivých klientských stanic. V dnešní době je snaha o co největší centralizaci zabezpečení a tím umožnit efektivnější správu. Bohužel, i když se NAT nemá považovat za bezpečnostní prvek, tak již nemožnost zvenčí adresovat stanice v síti poskytovala jistý druh ochrany. Což s použitím IPv6 odpadá. Další bezpečnostní prvek jako

jsou firewally budou mít svoji práci náročnější a to nejen díky masivní využití IPsec. Nový protokol tedy sebou přináší mnohé výhody, ale i mnoho nových otázek ohledně jeho provozu a dopadu na celkovou bezpečnost uzlů v síti IPv6 obsažených.

7 Závěr

V práci je velmi detailně popsána problematika týkající se přechodu od IPv4 na protokol IPv6. Uvedl jsem zde důvody, proč k přechodu musí dojít a co bude znamenat takový přechod pro běžného uživatele, tak i pro síťového administrátora.

V další části jsem se věnoval obecným principům zabezpečení a způsobům jak jim dostát. Poté jsem porovnal jednotlivé metody sloužící pro ochranu dat. Ty jsem se snažil porovnat se zabezpečením IPsec a tak utvořit obraz o možnostech, které máme k dispozici jako konkurenční řešení.

V části věnované IPsec jsem se snažil detailně rozebrat fungování celého protokolu spolu s jeho nabízenými službami. Díky tomu jsem dobře pochopil, jakým způsobem funguje výměna klíčů při použití Diffie-Hellman algoritmu. Způsob autentizace, šifrování a rozdíl při použití AH nebo ESP protokolu. Vybrané módy přenosu a šifrování jsem měl možnost poté otestovat v různých operačních systémech. V mnoha případech jsem narazil na velmi špatnou podporu ze strany operačního systému. Například v systému Windows XP a Windows 2003 server jsem nebyl schopen zprovoznit IPsec v IPv6 spojení pomocí AH s využitím staré specifikace algoritmů.

Další z mích úkolů bylo vytvořit návody na zprovoznění šifrovaného spojení pod různými operačními systémy. Ty jsem vytvořil na základě svých zkušeností při provádění praktické části. Snažil jsem se je podat takovým

způsobem, aby byli použitelné i pro méně zkušené uživatele daného systému.

Všechny mnou provedené testy probíhaly jak v IPv4, tak IPv6. V budoucím zkoumání této oblasti by bylo zajímavé, rozšířit testování o Windows server 2008. Nebo o některé z komerčních řešení například od společnosti Cisco, které by mělo podle úrovně jejich produktů přinést velmi kvalitní řešení. I přes to, pokud bych v dnešní době chtěl zabezpečit komunikaci mezi dvěma stanicemi, volil bych jednoznačně jeden ze systému Unix. Mnou testovaný Linux Ubuntu 9.04 obsahuje podporu pro IPsec v jádru a to již od verze 2.5. Dále nabízí širokou základnu algoritmů. Dlouhá podpora a velmi podařená implementace se podepsaly jak na bezproblémovém chodu a jednoduché konfiguraci, tak na interoperabilitě přesahující rámec ostatních implementací v operačních systémech.

8 Použitá literatura

1. DORASWAMY, Naganand, HARKINS, Dan. *IPSec : The New Security Standard for the Internet, Intranets, and Virtual Private Networks, Second Edition*. Edited by Mary Sudul, Noreen Regina; Alexis Heydt-Long; Gail Cocker-Bogusz, Meg VanArsdale, Anthony Gemmellaro, Jerry Votta. 2nd edition. [United States of America] : Prentice-Hall, 2003. 260 s. ISBN 0-13-046189-X.

2. DOSTÁLEK, Libor, KABELOVÁ, Alena. *Velký průvodce protokolyTCP/IP a systémemDNS, 2.aktualizované vydání*. Ivo Magera; Pavel Drinka. 2. aktualiz. vyd. Brno : Computer Press, 2000. 426 s. ISBN 80-7226-323-4.

3. HOFFMAN, Paul. *Cryptographic Suites for IPsec* [online]. 1 Santa Cruz : The Internet Engineering Task Force, 2005 [cit. 2009-08-09]. Dostupný z WWW: <http://www.ietf.org/rfc/rfc4308.txt?number=4308>.

4. KAUFMAN, Charlie. *Internet Key Exchange (IKEv2) Protocol* [online]. 1 Redmond : The Internet Engineering Task Force, 2005 [cit. 2009-08-09]. Text v angličtině. Dostupný z WWW: <http://www.ietf.org/rfc/rfc4306.txt?number=4306>.

5. KÁRA, Michal. Tuneluji, tuneluješ, tunelujeme : přesměrování portů. *Tuneluji, tuneluješ, tunelujeme* [online]. 2003, 1 [cit. 2009-07-09].

6. KENT, Stephen, SEO, Karen. *Security Architecture for the Internet Protocol* [online]. 1 [United States of America] : The Internet Engineering Task Force, 2005 [cit. 2009-08-09]. Text v angličtině. Dostupný z WWW: <http://www.ietf.org/rfc/rfc4301.txt?number=4301>.

7. KENT, Stephen. *IP Authentication Header* [online]. 1 [United States of America] : The Internet Engineering Task Force, 2005 [cit. 2009-08-09]. Text v angličtině. Dostupný z WWW: <http://www.ietf.org/rfc/rfc4302.txt?number=4302>.

8. KENT, Stephen. *IP Encapsulating Security Payload (ESP)* [online]. 1 [United States of America] : The Internet Engineering Task Force, 2005 [cit. 2009-08-09]. Text v angličtině. Dostupný z WWW: <http://www.ietf.org/rfc/rfc4303.txt?number=4303>.

9. LUHOVÝ, Karel. *VPN : historie, definice a důvody budování* [online]. 1 [Česká republika] : 2003 [cit. 2009-08-09]. Text v češtině. Dostupný z WWW: <http://www.svetsiti.cz/view.asp?rubrika=Tutorialy&temaID=219&clanekID=220>.

10. MITTCHEL, Bradley. *Create New VPN Connections in Windows XP Step by Step* [online]. 1 [2003] [cit. 2009-08-09]. Dostupný z WWW: <http://compnetworking.about.com/od/windowsxpnetworking/ss/newvpnconnect.htm>.

11. POPOVICIU, Ciprian, LEVY-ABEGNOLI, Eric, GROSSETETE, Patrick. *Deploying IPv6 Networks*. Edited by John Kane; Louisa Adair. 1st edition. Indianapolis : Cisco Press, 2006. 672 s. ISBN 978-1-58705-210-1.

12. PUŽMANOVÁ, Rita. *Virtuální privátní sítě pro vzdálený přístup* [online]. 1 [Česká republika] : 2006 [cit. 2009-08-09]. Text v češtině. Dostupný z WWW: <http://www.dsl.cz/clanky-dsl/clanek-511/virtualni-privatni-site-pro-vzdaleny-pristup>.

13. SNADER, John C. *VPNs Illustrated : Tunnels, VPNs, and IPsec*. 1st edition. Upper Saddle River : Addison Wesley Professional, 2005. 480 s. ISBN 0-321-24544-X.

14. SPENNEBERG, Ralf. *IPsec HOWTO* [online]. 2003 , 3. 9. 2005 [cit. 2009-08-09]. Text v češtině. Dostupný z WWW: <http://www.ipsec-howto.org/ipsec-howto_cz.html>.

15. STRAPA, Pavel. *Internetový protokol IPv6*. 2. vyd. Praha : CZ.NIC, 2008. 357 s. Dostupný z WWW: <http://knihy.nic.cz/ipv6/>. ISBN 978-80-904248-0-7.

16. *Secure Shell* [online]. [Česká republika] : [2008] , Stránka byla naposledy editována 7. 7. 2009 [cit. 2009-07-09]. Text v češtině. Dostupný z WWW: <http://cs.wikipedia.org/wiki/Secure_Shell>.

17. *Secure Sockets Layer* [online]. [Česká republika] : [2008] , Stránka byla naposledy editována 26. 6. 2009 [cit. 2009-08-09]. Text v češtině. Dostupný z WWW: <http://cs.wikipedia.org/wiki/Secure_Sockets_Layer>.

18. *PPTP* [online]. [Česká republika] : [2008] , Stránka byla naposledy editována 26. 5. 2009 [cit. 2009-08-09]. Text v češtině. Dostupný z WWW: <http://cs.wikipedia.org/wiki/PPTP>.

19. *Internet Protocol* [online]. [Česká republika] : [2006] , Stránka byla naposledy editována 25. 6. 2009 [cit. 2009-08-09]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/IPv4>.

20. *IPv6* [online]. [2006] , Stránka byla naposledy editována 25. 7. 2009 [cit. 2009-08-09]. Text v češtině. Dostupný z WWW: <http://cs.wikipedia.org/wiki/IPv6>.

9 Přílohy

Příloha č. 1

Součástí přílohy 1 je CD médium obsahující tuto práci v elektronické podobě.

Příloha č. 2A

Ubuntu 9.04

Zprovoznění IPsec pomocí ipsec-tools

1) Prvním krokem je připojení do sítě IPv6 k tomu můžeme použít například Miredo nebo jinou z alternativ připojení. V našem případě využijeme adres ze privátního rozsahu IPv6.

2) Provedeme kontrolu, zda je Ipv6 aktivní příkazem ifconfig. Pokud uvidíme následující výpis u rozhraní loopback tak je vše v pořádku.

   lo Link encap:Místní smyčka

   inet adr:127.0.0.1 Maska 255.0.0.0

   inet6-adr: ::1/128 Rozsah:Počítač

V případě, že ve výpisu není zahrnuta adresa loopback je potřeba použít příkaz modprobe ipv6 pro spuštění modulu pro práci s IPv6.

3) Přiřadíme adresu IPv6 našemu rozhraní.

   ifconfig rozhraní add Ipv6adresa/délka prefixu

   v našem případě: sudo ifconfig dev0 add 2001:db8::101/32

4) Otestujeme spojení pomocí ping6 příkazu.

   ping6 -I dev0 2001:db8::102/32

5) V dalším kroku nainstalujeme Ipsec-tools. (nenachází se ve standardních balíčcích Ubuntu je nutný přístup k internetu).

   sudo apt-get install ipsec-tools

6) V případě úspěšného dokončení instalace provedeme editaci souboru na adrese /etc/ipsec-tools.conf, kde nadefinujeme politiku, klíče a šifrovací algoritmy pro IPsec spojení.

```
# Konfigurace pro 2001:db8::101
# Vyčistí při startu SAD a SPD
flush;
spdflush;

# Nastavení AH SAs s použitím šifrování md5 a klíče o délce 128 bitů
#Výstupní politika pro AH
#-A označuje že se jedná o autentizaci
add 2001:db8::101 2001:db8::102 ah 0x200 -A hmac-md5
      0xc0291ff014dccdd03874d9e8e4cdf3e6;
# Vstupní politika pro AH
add 2001:db8::102 2001:db8::101 ah 0x300 -A hmac-md5
      0x96358c90783bbfa3d7b196ceabe0536b;

#Nastavení ESP SAs s použitým šifrováním 3des a klíčem o délce 192 bitů
#Výstupní politika pro ESP
#-E označuje, že se jedná o šifrování
add 2001:db8::101 2001:db8::102 esp 0x201 -E 3des-cbc
      0x7aeaca3f87d060a12f4a4487d5a5c3355920fae69a96c831;
#Vstupní politika pro ESP
add 2001:db8::102 2001:db8::101 esp 0x301 -E 3des-cbc
      0xf6ddb555acfd9d77b03ea3843f2653255afe8eb5573965df;

# Deklarace SA
# Povolení vstupu a výstupu z daných IP adres pro AH a ESP a specifikace
#přenosového režimu.
spdadd 2001:db8::101/32 2001:db8::102/32 any -P out ipsec
        esp/transport//require
        ah/transport//require;

spdadd 2001:db8::102/32 2001:db8::101/32 any -P in ipsec
        esp/transport//require
        ah/transport//require;
```

7) Po vložení předešlé konfigurace do souboru a následného uložení, provedeme vložení i na druhé straně tunelu pouze zaměníme politiky in a out u deklarace SA (nejednoduší způsob prohodit zmiňovaná slova). Dále můžeme načíst politiku do SAD.

   sudo /etc/init.d/setkey start

   popřípadě vyčistit nebo znovu načíst.

   sudo /etc/init.d/setkey stop

   sudo /etc/init.d/setkey restart

8) Pro případnou kontrolu můžeme použít příkaz **setkey -D**, který nám zobrazí aktuálně používanou politiku v SAD.

9) Tím jsme zprovoznili IPsec v transportní režimu a pro zabezpečení jsme použili AH k autentizaci a ESP pro šifrování.

10) Nyní otestujeme IPsec v tunelovém režimu se změnou v autentizaci. Rozdíl bude spočívat v použití ESP protokolu jak pro autentizaci tak pro šifrování. Celý postup je stejný jako v předchozím případě bod 6 až 8. Změny dosáhneme úpravou konfiguračního souboru ipsec-tools.conf podle následujícího předpisu. Nesmíme zapomenout při konfiguraci druhé strany tunelu také prohodit politiky in a out jako v případě transportního režimu.

```
# Konfigurace pro 2001:db8::101
# Vyčistí při startu SAD a SPD
flush;
spdflush;

#Nastavení ESP SAs s použitým šifrováním 3des a klíčem o délce 192 bitů
#autentizace provedená také díky ESP s použitím md5 o délce 128 bitů
#Výstupní politika
add 2001:db8::101 2001:db8::102 esp 0x201 -m tunnel -E 3des-cbc
0x7aeaca3f87d060a12f4a4487d5a5c3355920fae69a96c831
-A hmac-md5 0xc0291ff014dccdd03874d9e8e4cdf3e6;
#Vstupní politika
add 2001:db8::102 2001:db8::101 esp 0x301 -m tunnel -E 3des-cbc
0xf6ddb555acfd9d77b03ea3843f2653255afe8eb5573965df
-A hmac-md5 0x96358c90783bbfa3d7b196ceabe0536b;

# Deklarace SA
spdadd 2001:db8::101/32 2001:db8::102/32 any -P out ipsec
        esp/tunnel/2001:db8::101-2001:db8::102/require;

spdadd 2001:db8::102/32 2001:db8::101/32 any -P in ipsec
        esp/tunnel/2001:db8::102-2001:db8::101/require;
```

Příloha č. 2B

Windows XP

Zprovoznění IPsec pomocí podpory v systému

Provoz IPsec ve Windows XP je možný, ale jen za několika podmínek: použití omezeného množství šifrovacích algoritmů a nevyužití IPv6. Dále nesmíme zapomenout, že se jedná pouze o klienta takže není možné vytvořit spojení na protější Windows XP. K tomuto účelu Microsoft vyvíjí Windows server edition.

1) Start > Run > secpol.msc

2) Pravým tlačítkem klikneme na Zásady zabezpečení IP – Místní počítač a vybereme volbu vytvořit zásadu zabezpečení protokolu IP. V průvodci budete vyzváni k zadání jména politiky, mi zvolíme IPsec zabezpečení.

3) Deaktivujeme check box pro použití průvodce a vybereme volnu Přidat.

4) V nově otevřeném okně pojmenujeme pravidlo např. PC2LAN, vybereme možnost Přidat.

5) V dalším okně vybereme v poli Zdrojová adresa volbu adresa IP u tohoto počítače a pro cílovou adresu možnost Určená podsíť protokolů IP, kde vyplníme v našem případě hodnoty 192.168.1.0 255.255.255.0.

6) Odsouhlasíme okna, až k oknu Nové Pravidlo zde opět vybereme možnost přidat.

7) Nové pravidlo pojmenujeme LAN2PC a přidáme. V nově otevřeném okně doplníme hodnoty pro pole zdrojovou adresu vyberme možnost určená podsíť protokolů IP kde vyplníme v našem případě hodnoty 192.168.1.0 255.255.255.0. V poli cílová adresa zvolíme adresa IP u tohoto počítače.

8) Opět odsouhlasíme až k oknu Nové pravidlo. Zde označíme kolečkem pravidlo, které chceme editovat. Mi vybereme PC2LAN a přejdeme na kartu Akce Filtru a zakroužkujeme Požadovat zabezpečení (nepovinné). Klikneme na upravit a vybereme šifrovací algoritmu. Použijeme autentizaci a šifrování pouze za pomocí ESP s algoritmy 3Des a SHA1. Tuto kombinaci posuneme na první pozici. Pomocí check boxu necháme aktivní pouze volbu Metoda Perfec Forward secrecy pro klíč relace.

9) V záložce Metody ověřování vyberme před sdílený klíč a vložíme hodnotu například „XYZ12345“.

10) Přepneme na záložku Nastavení tunelového propojení a zadáme jako konec tunelu adresu protějšího počítače 192.168.1.2

11) V poslední záložce Typ připojení zvolíme volbu Všechna síťová připojení a vše potvrdíme OK.

12) Nyní jsme se vrátili opět do okna s názvem našich politik, kde se nám nově objevila položka PC2LAN. Zde opět vybereme možnost Přidat.

13) A opakujeme celý proces od bodu 8 s dvěma rozdíly. První je že vyberme na začátku kolečkem možnost LAN2PC. Druhý rozdíl se nachází v záložce Nastavení tunelového propojení, kde zadáme adresu místní stanice.

14) Po provedení nastavení LAN2PC se nacházíme opět v okně s názvem naší politiky. Zde zkontrolujeme jestli je check box u námi vytvořených politik zaškrtnutý a odsouhlasíme kliknutím na OK.

15) Posledním krokem je kliknutím pravým tlačítkem na naši politiku ve sloupci Zásady jsou přiřazeny a vybrat možnost Přidělit. Tím jsme uvedli naši politiku v platnost a nyní můžeme navázat bezpečnou komunikaci například příkazem ping 192.168.1.2. V případě užití ping bude prvních 5 paketů využito pro vyjednání politiky pak, už již přijde na řadu zabezpečený tunel.

    Zde popsaná konfigurace odpovídá s drobnými rozdíly i konfiguraci na Windows server 2003. Také se dá použít nejen oproti stanici s Windows, ale ke spojení s vzdáleným routerem.

Příloha č. 2C

Windows Vista

Zprovoznění IPsec pomocí podpory v systému

IPsec v systému Windows Vista je již na velmi dobré úrovni a jsou zde implementované nástroje i na zrychlení jeho konfigurace. První způsob jak nakonfigurovat parametry pro zabezpečený tunel je shodný se systémem Windows XP. Takže pomocí secpol.msc nadefinovat politiku pro připojení. Další variantou je využít integrovaného nástroje v podobě Brány firewall s vyspělým zabezpečením. Postup konfigurace je následující.

1) Nejdříve spustíme pomocí Start > spustit > cmd.

2) Necháme si vypsat indexy používané pro všechna síťová rozhraní s podporou IPv6.

   netsh interface ipv6 show interface

3) Vybereme námi používané rozhraní a následujícím příkazem mu přiřadíme naši IPv6 adresu.

   netsh interface ipv6 add address adresa_rozraní

   v našem případě bude vypadat příkaz následovně

   netsh interface ipv6 add address 2001:db8::101

4) Skrz z nabídky Start vybereme Nástroje pro správu a spustíme konfigurační program Brána firewall s vyspělým zabezpečením.

5) Z Levého sloupce kliknutím zobrazíme položku Pravidla zabezpečení připojení.

6) V pravém sloupci pojmenovaném jako Akce zvolíme možnost Nové Pravidlo

7) Nyní ve zobrazeném průvodci vybereme možnost Tunel a klikneme na Další.

8) U pole Počítače na konci 1 klikneme na tlačítko Přidat a zapíšeme IP adresu 2001:db8::101

9) V dalším poli Místní počítač tunelového propojení zapíšeme adresu IPv6 a to 2001:db8::101

10) Do pole Vzdálený počítač tunelového propojení vložíme adresu IPv6 a to 2001:db8::102

11) Tuto adresu také vložíme do pole Počítače na konci pomocí tlačítka Přidat.

12) V dalším kroku zvolíme Před sdílený klíč a vložíme sem hodnotu XYZ12345.

13) Po kliknutí na další budeme dotázáni na situaci kdy má být pravidlo použito. V našem případě necháme zaškrtnuté všechny možnosti.

14) Nyní již následuje pouze dotaz na jméno našeho pravidla a odsouhlasením pomocí tlačítka OK je vše nastaveno.