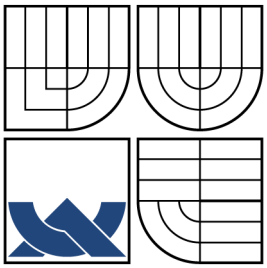
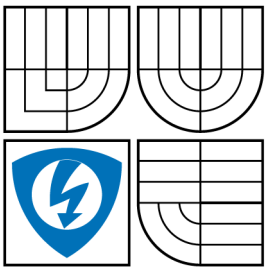


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND
COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

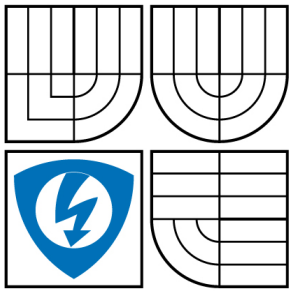
ZABEZPEČENÝ PEER TO PEER KOMUNIKAČNÍ SYSTÉM
SECURE PEER-TO-PEER COMMUNICATION SYSTEM

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — BC. LUBOŠ ELIÁŠ
AUTHOR

VEDOUCÍ PRÁCE — ING. LUBOMÍR CVRK, PH.D.
SUPERVISOR

BRNO 2008



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky a komunikačních technologií

Ústav telekomunikací

Diplomová práce

magisterský navazující studijní obor

Telekomunikační a informační technika

Student: Eliáš Luboš Bc. ***ID:*** 88523

Ročník: 2 ***Akademický rok:*** 2007/2008

NÁZEV TÉMATU:

Zabezpečený peer to peer komunikační systém

POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je implementovat obecný zabezpečený peer-to-peer komunikační systém. Je nutné implementovat navázání spojení tak, aby oba, jeden, nebo žádný z komunikujících mohli být ukryti za překladem síťových adres. K navázání spojení implementujte spojovací server. Komunikační tunel musí být chráněn šifrováním a probíhat přes jediný UDP port. V tomto tunelu musí být možné přenášet data libovolných jiných síťových služeb. Je nutné nastudovat principy tzv. UDP hole punching, asymetrické a symetrické kryptografie. Systém implementuje v prostředí MS Windows XP v programovacím jazyku C++.

DOPORUČENÁ LITERATURA:

[1] FERGUSSON, N., SCHNEIER, B. Practical Cryptography. Wiley Publishing, Inc., Indianopolis USA, 2003.
[2] SENIE, D. Network Address Translator (NAT)-Friendly Application Design Guidelines. RFC3235, Network Working Group, 2002.
[3] KEGEL, D. NAT and Peer-to-peer networking, Webová stránka http://alumnus.caltech.edu/~dank/peer-nat.html, 1999.
[4] CVRK, L. Kryptografické metody zabezpečení datových přenosů, Disertační práce, 2006.

Termín zadání: 11.2.2008 ***Termín odevzdání:*** 28.5.2008

Vedoucí práce: Ing. Lubomír Cvrk, Ph.D.

prof. Ing. Kamil Vrba, CSc.
předseda oborové rady

UPOZORNĚNÍ:

LICENČNÍ SMLOUVA

POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami:

1. Pan/paní

Jméno a příjmení: Bc. Luboš Eliáš

Bytem:

Narozen/a (datum a místo): 11.1.1984, Třebíč

(dále jen "autor")

a

2. Vysoké učení technické v Brně

Fakulta elektrotechniky a komunikačních technologií

se sídlem Údolní 244/53, 60200 Brno 2

jejímž jménem jedná na základě písemného pověření děkanem fakulty:

prof. Ing. Kamil Vrba, CSc.

(dále jen "nabyvatel")

Článek 1
Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):
   - ☐ disertační práce
   - ☑ diplomová práce
   - ☐ bakalářská práce

   jiná práce, jejíž druh je specifikován jako .....

(dále jen VŠKP nebo dílo)

Název VŠKP: Zabezpečený peer to peer komunikační systém

Vedoucí/školitel VŠKP: Ing. Lubomír Cvrk, Ph.D.

Ústav: Ústav telekomunikací

Datum obhajoby VŠKP: .....

VŠKP odevzdal autor nabyvateli v:

- ☑ tištěné formě - počet exemplářů 1
- ☑ elektronické formě - počet exemplářů 1

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

Článek 2
Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizovaní výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti
   - ☑ ihned po uzavření této smlouvy
   - ☐ 1 rok po uzavření této smlouvy
   - ☐ 3 roky po uzavření této smlouvy
   - ☐ 5 let po uzavření této smlouvy
   - ☐ 10 let po uzavření této smlouvy

   (z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

Článek 3
Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísni a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne: .....

....

Nabyvatel Autor

ABSTRAKT

Hlavním cílem této diplomové práce je implementace obecného zabezpečeného peer-to-peer komunikačního systému. Systém má schopnost zcela automatizovaně navázat a provozovat zabezpečené spojení mezi koncovými uzly komunikace. Tuto schopnost má i v případě, kdy se v cestě k cílovému systému nachází překladače síťových adres, bez nutnosti explicitně tyto překladače konfigurovat. Zabezpečovací procedury tohoto systému jsou transparentně skryty před jednotlivými síťovými aplikacemi, které tento problém musely řešit každá samostatně. Odpovědnost za bezpečnost je přenesena na aplikačně nezávislý subsystém pracující v rámci jádra operačního systému. Bezpečnost tohoto subsystému je založena na zachytávání odchozích a příchozích IP paketů a jejich autentizaci a šifrování. Systém se podařilo implementovat na operačním systému Microsoft Windows XP, v programovacím jazyku C++. Byla změřena přenosová rychlost komunikačního tunelu systému na různých šířkách pásma sítě. Výsledky ukazují, že při použití systému na běžných dnešních počítačích nedochází k prakticky žádnému zpomalení přenosové rychlosti oproti nešifrovanému kanálu.

KLÍČOVÁ SLOVA

komunikace, kryptografie, peer to peer, NAT, UDP hole punching, síťové programování, C++, MS Windows, IPsec, SSL/TLS

ABSTRACT

The main aim of this master's thesis is to implement a common, secure and peer-to-peer communication system. The system has ability to automatically establish and run a secure end-to-end connection. It has this ability even if a network address translator is in the way to the destination system, without need of any explicit configuration of this translator. The security procedures of this system are in a transparent manner masked from individual applications, which had to solve this challenge in their own way. A responsibility for a security is delegate to an application-independent subsystem working within the core of an operating system. The security of this subsystem is based on capturing the outbound and inbound IP packets and their authentication and encryption. The system was successfully implemented in MS Windows XP operating system, in programming language C++. Transfer rate of communication tunnel in different network bandwidth speeds was measured. Result shows, that in the case of use the system on standard PC sold nowadays is practically no decrease of the transfer rate in comparison to a common channel.

KEYWORDS

communication, cryptography, peer-to-peer, NAT, UDP hole punching, network programming, C++, MS Windows, IPsec, SSL/TLS

ELIÁŠ, L. *Zabezpečený peer to peer komunikační systém*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2008. Počet stran 64, počet stran příloh 1. Vedoucí diplomové práce byl Ing. Lubomír Cvrk, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Zabezpečený peer to peer komunikační systém“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne ....

(podpis autora)

PODĚKOVÁNÍ

Děkuji vedoucímu diplomové práce, Ing. Lubomíru Cvrkovi, Ph.D, za velmi užitečnou pomoc a nadmíru cenné rady při zpracování této diplomové práce. Dále děkuji spolužákům Bc. Miroslavu Greplovi a Bc. Václavu Dohnálkovi za poskytnutou pomoc při testování částí vytvářeného systému.

V Brně dne ....

(podpis autora)

OBSAH

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

ÚVOD

Tato diplomová práce se zabývá zabezpečením datových přenosů v sítích založených na protokolu IP [5]. Bude realizovat v praxi systém založený na teorii tzv. implicitní bezpečnosti datových toků. Teorie implicitní bezpečnost je založena na myšlence zabezpečit všechnu datovou komunikaci mezi koncovými uzly v síti, které ji podporují, ale přitom umožnit i komunikaci nezabezpečenou mezi uzly, které ji nepodporují, a to pořád s podporou stejných služeb, které by běželi i bez implicitní bezpečnosti. Slovo implicitní znamená, že uživatelé ani aplikace se po správném nastavení nebudou muset již o zabezpečení starat. Uživatel poskytne systému pouze svůj privátní klíč, a bude schvalovat, ke kterým systémům se připojit. Zabezpečený bude celý datový tok mezi koncovými uzly sítě a to i v případech, kdy se na cestě k cílovému systému nachází překladače síťových adres. Tato síťová zařízení je při navazování spojení mezi uzly velmi obtížné obejít. U zabezpečovacích systémů používaných v současnosti, je bezpečné spojení mezi koncovými uzly zakončeno na serveru VPN. Datové toky mezi serverem VPN a koncovým uzlem v rámci jeho privátní sítě jsou nezabezpečené. Pokud navíc server VPN leží až za překladačem síťových adres, vzdálený uzel se k němu nepřipojí, pokud nebude překladač takto speciálně nakonfigurován a nebude tedy propouštět všechny pakety k serveru VPN.

Cílem práce je implementovat obecný zabezpečený peer-to-peer komunikační systém. Tento systém byl navržený v rámci disertační práce [1]. Z této práce je také převzat teoretický popis systému uvedený v úvodních kapitolách. Systém má být implementován v prostředí MS Windows XP, v programovacím jazyku C++. Je nutné vyřešit navázání spojení tak, aby oba, jeden, nebo žádná z komunikujících stran mohla být ukryta za překladem síťových adres. Řešení tohoto problému poskytuje technologie UDP hole punching[1]. K navázání spojení je potřeba implementovat spojovací server. Komunikační tunel tohoto systému musí být chráněn šifrováním s využitím asymetrické a symetrické kryptografie a probíhat přes jediný UDP port. V tomto tunelu musí být možné přenášet data libovolných jiných síťových služeb.

Obecné principy systémy jsou popsány v kapitolách 2 a 3. Popisu klienta se věnuje více kapitola 5 a popisu spojovacího serveru kapitola 4. Mechanismus navazování spojení rozebírá kapitola 6 a mechanismus zabezpečení datových toků kapitola 7. Testování korektního navazování spojení a sestavení komunikačního tunelu, při různých síťových umístění klientů, je ukázáno v kapitole 8. V této kapitole jsou také provedeny testy rychlosti tunelu systému a srovnány s rychlostmi dosaženými sys-

[1] Nepodporují všechny typy překladačů síťových adres.

témy IPsec a OpenVPN při podobném zabezpečení komunikačního tunelu.

1 SOUČASNÉ SYSTÉMY ŘEŠÍCÍ ZABEZPEČENÍ DATOVÝCH PŘENOSŮ

V nynější době jsou používány zejména dva obecné přístupy pro ochranu datových přenosů. Jde o standard SSL/TLS [9] a o standard IPsec [10]. Na těchto standardech jsou postavena řešení jako např. FreeS/WAN [12] (IPsec), OpenVPN [16] (SSL/TLS VPN) a mnoho podobných dalších.

1.1 IPsec, IPsec VPN

[3] Řešení VPN využívajících protokolu IP (IP VPN) je pro zabezpečení nejčastěji založeno na IPsec. IPsec označuje několik protokolů na síťové vrstvě. Ty nám jako celek poskytují možnost tunelování, šifrování a autentizaci. Uzly, které spolu chtějí komunikovat s využitím IPsec, se musí nejprve dohodnout na bezpečnostní politice. Tunel mezi dvěma servery nebo mezi serverem a uživatelem pak zabezpečuje provoz jakéhokoli typu.

IPsec je pro budování VPN z hlediska bezpečnosti metoda velmi bezpečná. Bohužel je ale současně také velmi složitou metodou na implementaci. Jeho hlavní nevýhodu lze spatřovat v nutnosti komplexní konfigurace všech potřebných parametrů. IPsec také může mít potíže v prostředí s překladem síťových adres.

Autentizace v rámci IPsec probíhá podle toho, zda se používá režim tunelovací, nebo transportní. Transportní režim se používá tehdy, pokud obě komunikující strany podporují IPsec. Záhlaví typu AH nebo ESP se vkládá za původní záhlaví paketu a šifrují se pouze data, což je výhodné z hlediska zachované podpory pro QoS (Quality of Service). Tunelovací režim vkládá celý paket s příslušným záhlavím (AH nebo ESP) do nového paketu, takže šifrování probíhá přes celý původní paket. Režim tunelu umožňuje síťovým zařízením, jako směrovačům, pracovat jako IPsec servery pro více uživatelů VPN. Tunelovací režim ESP je nejčastěji používaný pro budování VPN založených na protokolu IP.

1.2 SSL/TLS, SSL VPN

[3] SSL/TLS (SSL) je využívána jako alternativa k IPsec. Pracuj ovšem až na aplikační vrstvě, je tedy pro budování VPN slabší z hlediska bezpečnosti, zabezpečuje totiž pouze některé aplikace. Výhodou je, že je méně náročná na implementaci.

SSL zajišťuje autenticitu odesílatele, integritu a šifrování aplikačních dat při jejich přenosu přes veřejnou IP síť. SSL vyžaduje spolehlivý protokol na transportní vrtsvě.

SSL podporuje obousměrnou autentizaci, ale používá se většinou pouze jednosměrně. Lze využít jak jména a hesla (RADIUS), jména a token (RSA SecureID), nebo digitální certifikáty X.509. Klient SSL spoléhá na digitální certifikát od vzdáleného serveru, k němuž se chce připojit.

1.3 Srovnání SSL/TLS a IPsec

[3] Porovnání IPsec s alternativním SSL pro budování VPN je uvedeno v tab. 1.1. IPsec se stará o bezpečný komunikační kanál po IP mezi klientem a cílovým systémem na síťové vrstvě, takže nijak neomezuje podporované aplikace. Naproti tomu SSL slouží k zabezpečení některých aplikací typu klient-server, všechny aplikace pak v sobě musí mít implementováno SSL.

Úroveň SSL šifrování se dohaduje při komunikaci tak, že se většinou strany shodnou na nejnižším možném stupni, tedy na 40bitovém klíči. U IPsec je úroveň šifrování přednastavená administrativně, takže běžně lze použít 128bitový klíč. SSL šifruje nad transportní vrstvou, takže chrání aplikační data. IPsec, s použitím protokolu ESP, chrání šifrováním navíc původní záhlaví IP a TCP. IPsec rovněž chrání před nechtěnou analýzou provozu, a to SSL neumí. SSL není chráněno ani před některými útoky typu DoS (Denial of Service).

Jak SSL tak IPsec podporuje bezpečnostní algoritmy pro zajištění integrity zpráv, jako MD5 nebo SHA1, ale implementace IPsec podporují silné šifrování typu 3DES nebo AES (Advanced Encryption Standard).

SSL se hodí i pro připojování mobilních uživatelů z veřejných přístupových sítí (např. bezdrátových LAN, tzv. hot spots), IPsec naproti tomu je vhodné pro pevné připojení vzdálených uživatelů.

Tab. 1.1: Porovnání VPN na bázi IPsec a SSL [3]

	SSL	**IPsec**
aplikace	webové aplikace, sdílení souborů a elektronická pošta	všechny aplikace TCP/IP
šifrování	různě silné v závislosti na webovém prohlížeči	konzistentně silné, závisí na dané implementace
autentizace	různě silná (jednosměrná nebo obousměrná, za použití hesel, tokenu nebo certifikátů)	silná (obousměrná, za použití tokenu nebo certifikátů)
bezpečnost celkově	středně silná	velmi silná

2 DECENTRALIZOVANÝ ZABEZPEČENÝ KOMUNIKAČNÍ SYSTÉM

Systém zabezpečení datových toků navržený v rámci disertační práce [1] je založen na kryptografickém subsystému, který je nezávislý na jakékoliv aplikaci. Cílem bylo, aby pro síťové aplikace byl zabezpečovací postup transparentní, aby se o něj nemuseli vůbec starat. Výhodou v takovém případě je jistě úspora času při vývoji síťových aplikací, kdy není nutné zabezpečení vůbec implementovat.

Navržený systém tedy přesunul zabezpečovací proces komunikace do síťové vrstvy ISO/OSI modelu. Všechny aplikační síťové programy totiž musí nechat svá data poslat do sítě přes operační systém prostřednictvím síťové vrstvy.

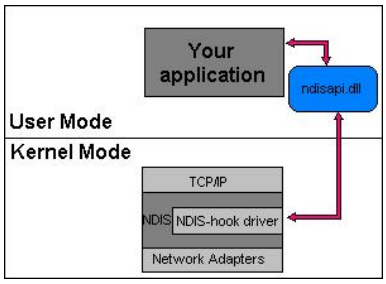
Síťová (transportní i linková) vrstva je implementována v operačním systému, proto možnosti ke kontrole a řízení datového přenosu jsou omezenější a vyžadují znalosti jádra operačního systému. Je totiž nutné implementovat mechanismus, který před odesláním rámce do fyzické vrstvy rámec zadrží, provede nad ním požadovanou transformaci a poté jej odešle. Pro přijatá data je požadovaná funkcionalita přesně opačná, tedy pro každý rámec přijatý z fyzické vrstvy musí být před jeho předání vyšším vrstvám provedena opět požadovanou transformaci.

2.1 Paketový procesor

Aby bylo možné zachytit a modifikovat pakety, je nutné do operačního systému implementovat ovladač, který ovlivňuje činnost tzv. protokolového zásobníku (protocol stack).

Pro tyto účely bylo použito na operačním systému Microsoft Windows XP vývojové prostředí WinpkFilter [18]. Prostřední obsahuje kernel-level NDIS-hook ovladač v programovacím jazyku C, který umožňuje práci s daty v rámci jádra operačního systému a v rámci tzv. user-mode. Kromě toho je také možné pracovat s daty v user-mode, a to prostřednictvím rozhraní mezi jádrem a uživatelským režimem. Toto je zobrazeno na obr. 2.1. Knihovna ndisapi.dll realizuje toto rozhraní mezi jádrem a uživatelským režimem.

Paketový procesor je založen na uvedeném mechanismu zachytávače paketů. Zasílá požadavky tzv. *spojovací službě* na vyjednávání klíčů pro symetrické šifry, hlídá žádosti o zahájení spojení, vytváří zabezpečený komunikační tunel a realizuje kryptografické zabezpečovací operace.



Obr. 2.1: Komunikační model zachytávače paketů mezi kernel-mode a user-mode [18]

2.2 Decentralizovaný model komunikace

Zabezpečený komunikační systém, navržený v rámci disertační práce [1], byl pojmenován DETEC[1] – zkratka z anglického spojení Decentralized End-To-End Communicator, což lze přeložit jako Decentralizovaný systém pro komunikaci mezi koncovými body. Základem návrhu komunikační architektury systému byla nutnost kryptograficky zabezpečit komunikaci mezi koncovými klienty, která dosud nebyla uspokojivě řešena. Typicky používaná řešení založená na VPN zabezpečí komunikaci od klienta k VPN serveru, který ale není totožný s klientem, se kterým chceme komunikovat. Proto lze na cílové lokální síti nalézt část, přes kterou procházejí pakety nezabezpečené a proto je možné odposlouchávat komunikaci. Aby bylo toto omezení možné obejít a komunikovat zabezpečeně mezi koncovými systémy, je nezbytně nutné umět navázat přímé spojení mezi koncovými body, i pokud se, v nejhorším případě, oba nacházejí v privátních sítích za překladem adres.

2.3 UDP hole punching

Technika UDP hole punching [2] umožňuje založení přímého (peer-to-peer) spojení mezi dvěma klienty, za pomoci tzv. setkávacího serveru, i pokud jsou oba za NATem.

2.3.1 Server

UDP hole punching předpokládá, že oba klienti, *A* a *B*, mají již aktivní UDP spojení se serverem *S*. Jakmile se klient připojí na server, server si uloží IP adresu a UDP port obou jeho koncových bodů: privátního koncového bodu a veřejného koncového bodu. Privátní koncový bod je bod v rámci privátní sítě klienta, a veřejný bod vidí server po překladu privátního NATem. Server získá privátní koncový bod z klientovy

[1] čti: [di:tek]

registrační zprávy, a veřejný získá z IP a UDP hlavičky této zprávy. Pokud klient není za NATem, budou tedy oba koncové body identické.

Některé druhy NATů hledají v těle UDP paketu 4 B pole, které vypadá jako IP adresa a přeloží ji stejně jako privátní IP adresu. Tento problém není třeba řešit u aplikací, které data šifrují.

2.3.2 Založení peer-to-peer spojení

Předpokládejme, že klient *A* chce založit UDP spojení s klientem *B*. UDP hole punching tedy bude vypadat takto:

1. *A* nejprve neví jako kontaktovat *B*, tak požádá *S* o pomoc se založením UDP spojení s *B*.

2. *S* odpoví *A* zprávou obsahující veřejný a privátní body *B*. Ve stejné době použije své UDP spojení s *B* a pošle mu zprávu obsahující požadavek na spojení od *A* a v ní koncové body *A*. *A* i *B* tak nyní znají navzájem oba koncové body sousední strany.

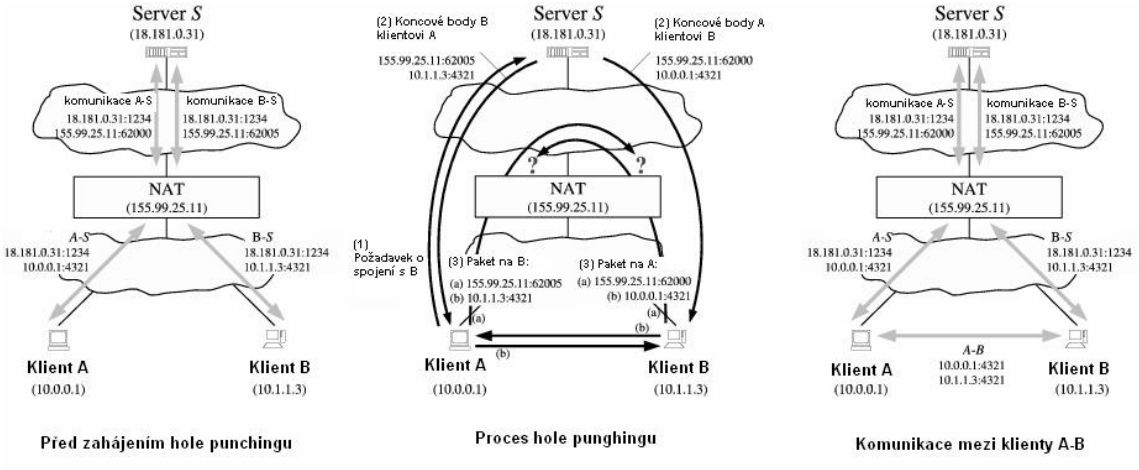
3. Když *A* přijme koncové body *B* od *S*, tak začne na oba tyto koncové body posílat UDP pakety a sleduje, ze kterého nejdříve dostane odpověď od *B*. Stejné provede i obráceně *B*. Není rozhodující jestli začne dříve *A*, nebo *B*.

UDP hole punching musí pracovat v různých síťových scénářích. V prvním, nejjednodušším, jsou oba klienti za stejným NATem, v jedné privátní síti. Ve druhém, nejčastěji se vyskytujícím, jsou oba klienti za různými NATy. Ve třetím případě jsou oba klienti za více úrovněmi NAT.

2.3.3 Klienti za společným NAT

Tento scénář je zobrazen na obr. 2.2. Klient *A* sestavil UDP spojení se serverem *S*. NAT přeložil číslo UDP portu na 62000. Klient *B* udělal totéž a stejný NAT přeložil číslo UDP portu na 62005.

A nyní využije techniky hole punchingu k sestavení UDP spojení s *B*, prostřednictvím serveru *S*. *A* pošle *S* zprávu obsahující žádost o spojení s *B*. *S* pošle *A* oba koncové body *B*, a *B* naopak koncové body *A*. Oba klienti se nyní pokusí posílat UDP pakety sobě navzájem na oba koncové body. Pakety poslané na veřejné koncové body buď dorazí, nebo ne. To záleží na tom, jestli NAT podporuje Hairpin překlad 2.3.6. Pakety směrované na privátní koncové body dosáhnou cíle. Pokud dorazí pakety směrované na oba koncové body, je kvůli výkonu lepší vybrat privátní koncové body.

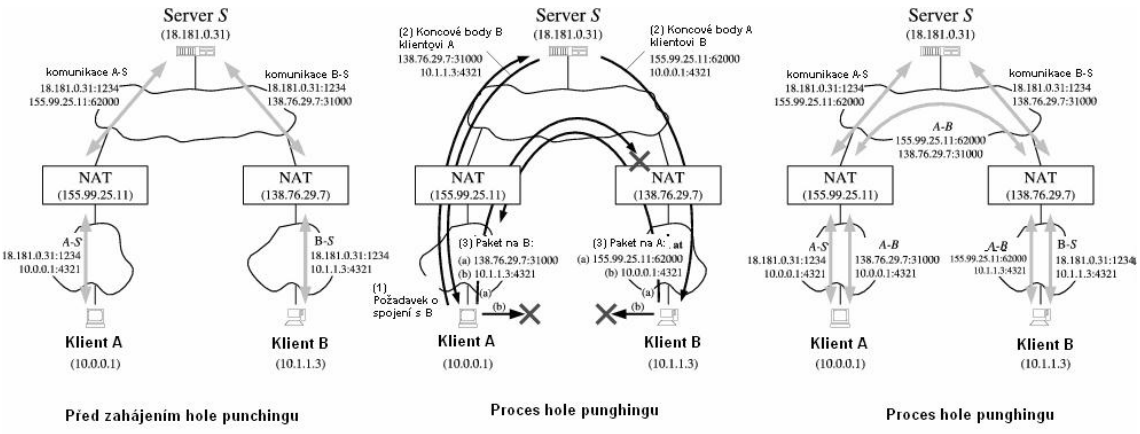


Obr. 2.2: UDP hole punching, klienti za společným NAT [1]

2.3.4 Klienti za různými NAT

Tento scénář je zobrazen na obr. 2.3. A a B mají navázáno spojení ze svého lokálního portu 4321 na port 1234 serveru S. NAT A přiřadí pro toto spojení IP:port 155.99.25.11:62000. NAT B přiřadí klientu B 138.76.29.7:31000.

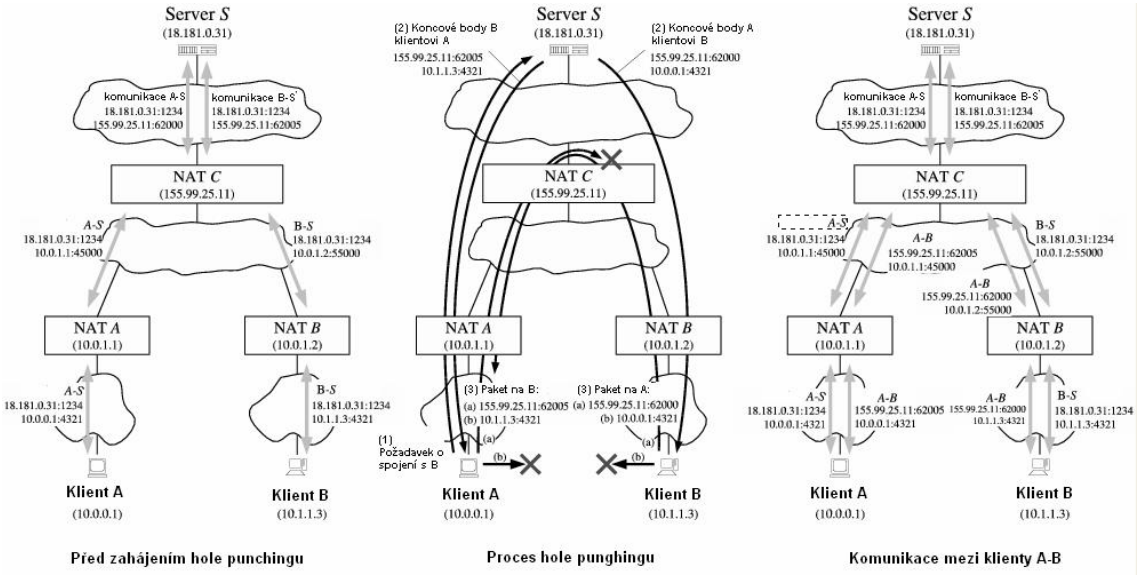
V registrační zprávě A pro S pošle A své privátní koncové body S jako 10.0.0.1:4321. S si uloží oba dva koncové body. Veřejný koncový bod A je v tomto případě je 155.99.25.11:62000. Stejně se server S dozví koncové body od B. V našem případě je privátní 10.1.1.3:4321 a veřejný 138.76.29.7:31000. Nyní může pokračovat proces UDP hole punching tak jak je popsán výše. Klienti budou na konci procesu komunikovat přes jejich veřejné koncové body, protože privátní nemůžou být použitelné.



Obr. 2.3: UDP hole punching, klienti za různými NAT [1]

2.3.5 Klienti za více úrovněmi NAT

Tento scénář je zobrazen na obr. 2.4. Můžeme si představit, že NAT C je provozovaný poskytovatelem připojení k Internetu a NATy A a B jsou zákaznické routery domácích sítí. Veřejné IP adresy mají tedy pouze server *S* a NAT C. NATy A a B mají IP adresy přidělené poskytovatelem připojení a klienti mají adresy vnitřní sítě. Klienti založí jako dříve spojení se serverem *S*. Při tomto spojení dojde k dvojnásobnému překladu adres, první na NATech A i B a druhý na NATu C. Klienti v tomto případě nemají na výběr a musí využít veřejných koncových bodů, pod kterými je vidí server. NAT C musí podporovat hairpin překlad síťových adres 2.3.6. Když klient *A* pošle UDP paket klientu *B* na jeho veřejný koncový bod 155.99.25.11:62005, nejdříve dojde k překladu na NATu A z 10.0.0.1:4321 na 10.0.1.1:45000. Poté paket dorazí na NAT C, který by měl poznat, že cílová adresa je jeden z jeho vlastních přeložených veřejných koncových bodů. Pokud má NAT C podporu takového překladu, pak paket vezme, přeloží jeho zdrojovou a koncovou adresu a pošle paket zpět do sítě.



Obr. 2.4: UDP hole punching, klienti za více úrovněmi NAT [1]

2.3.6 Hairpin překlad

Některé síťové konfigurace vyžadují podporu hairpin překladu u zařízení provozujících NAT službu, aby mohl UDP hole punching fungovat. Na obr. 2.4 např. musí NAT C tento překlad podporovat. Před třemi roky v době publikování [2] byla podpora tohoto překladu u NATů relativně malá. Mnoho síťových konfigurací ale

naneštěstí tento překlad vyžaduje, proto je potřeba tento překlad implementovat v budoucích implementacích NATů.

2.3.7 Zrušení NAT pravidel pro UDP pakety

Po odeslání UDP paketu a vytvoření pravidla na NATu pro průchod paketu obsahujícího odpověď zpět je počítán čas existence tohoto pravidla. Není totiž jisté, zda UDP paket odpovědi bude vůbec odeslán zpět. Časový interval existence pravidla se ale liší NAT od NATu. Doba existence tohoto pravidla se pohybuje okolo 20 s. Proto je třeba odesílat při potřebě zachovat spojení odesílat pakety, které udrží spojení, popř. bude nutné znovu provést proces UDP hole punching.

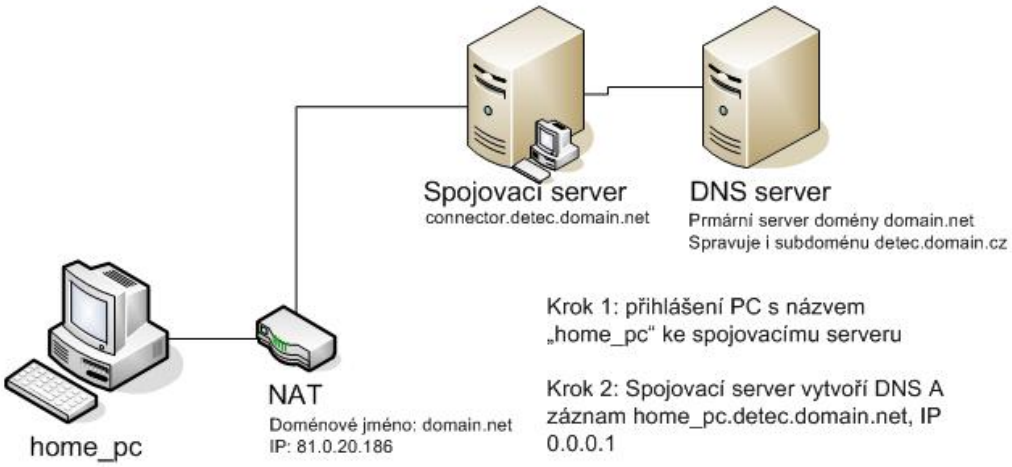
3 PRINCIPY SYSTÉMU

Aby bylo možné docílit plnohodnotné nezávislosti aplikace na subsystému zabezpečení datových toků, nesmí být požadována jakákoliv modifikace aplikace, aby se sestavilo a provozovalo zabezpečené spojení.

Každá žádost o spojení v libovolné aplikaci, ať už jde o webový prohlížeč, správce souborů apod., začíná vždy DNS dotazem, neboť je nutné přeložit doménové jméno cíle, se kterým je třeba se spojit (např. www.seznam.cz) na odpovídající IP adresu. Tohoto faktu je využito při realizaci automatického navazování spojení. Systému DNS je tak využito pro označení klientů, kteří se nacházejí ukryti za překladačem síťových adres.

3.1 Přihlášení do systému

Systém DETEC úzce spolupracuje s primárním DNS serverem spravujícím záznamy domény, která zastupuje veřejnou adresu sítě skryté za překladačem síťových adres. Klient systému se nejprve musí připojit ke spojovacímu serveru, kooperujícímu se zmíněným DNS serverem. Připojení klienta do systému DETEC je zobrazeno na obr. 3.1.



Obr. 3.1: Připojení klienta do systému DETEC [1]

Při spouštění tzv. registrační služby systému DETEC na počítači home-pc (viz 5.1), se připojí počítač nejprve ke spojovacímu serveru, který je přiřazen systému DETEC pro danou doménu. V příkladu na obrázku je uveden překladač síťových adres s veřejnou adresou, které je přiřazeno doménové jméno domain.net. Pro doménu domain.net musí ve veřejné síti existovat spojovací server connector.detec.domain.net.

K tomuto serveru se klient přihlašuje. Tento server úzce spolupracuje s primárním DNS serverem spravujícím doménu detec.domain.net.

Spojovací server je oprávněn zakládat DNS záznamy pro doménu detec.domain.net. Při registraci je pro home-pc vytvořeno doménové jméno home-pc.detec.domain.net s IP adresou 0.0.0.1 (vysvětleno v odstavci 3.2). Pod uvedeným doménovým jménem se počítač home-pc stává přístupným z veřejné sítě za použití systému DETEC. Ten, kdo se chce spojit s počítačem home-pc musí znát toto doménové jméno. Z doménového jména odvodí přihlašovací subsystém název spojovacího serveru (connector.detec.domain.net). Příslušný spojovací server je tedy požádán o sestavení spojení s jedním z klientů náležejících do jeho domény.

V 1. kroku zašle klient svému spojovacímu serveru přihlašovací zprávu, kde je uvedena lokální IP adresa síťového rozhraní klientského počítače a MAC adresa síťového rozhraní. Server zjistí, zda má být danému klientu přiřazeno statické, nebo dynamické DNS jméno, podle konfigurace serveru. V dalším kroku je přidáno klientovo DNS jméno do DNS s IP adresou 0.0.0.1, pokud je klient za překladačem síťových adres, nebo jeho běžná adresa, pokud je ve veřejné síti. Spojovací server má oprávnění přidávat a mazat A záznamy v rámci domény detec.domain.net. Proces odhlášení je analogický přihlášení. Odešle se zpráva logout, kde je uvedena lokální IP a MAC adresa (stejně jako při přihlášení). Spojovací server zajistí odstranění příslušného A záznamu z DNS.

Systém DETEC přiřazuje všem klientům za překladači síťových adres IP adresu 0.0.0.1. Tato adresa je z adresního prostoru 0.0.0.0/8 - „tato síť“ (this network) rezervovaného organizací IANA. Vysvětlení přiřazení této adresy je v odstavci 3.2. Jak již bylo řečeno, klientům, kteří jsou ve veřejné síti, je do DNS záznamu vložena jejich skutečná adresa. Pokud je klient za překladačem síťových adres, udržuje si po celou dobu, co je připojen do systému DETEC, TCP spojení se svým spojovacím serverem, aby se s klientem přes spojovací server mohl spojit kdokoliv z veřejné sítě.

3.2 Otevření spojení

Spojení na další cílový systém, mimo těch, ke kterým již spojení existuje, začíná DNS dotazem gethostbyname(”home-pc.detec.domain.net”); Tato funkce socket API požádá DNS server, aby zjistil IP adresu daného doménového jména.

Zjišťování adresy pro počítač, registrovaný v danou chvíli v DNS pod jménem home-pc.detec.domain.net, probíhá následujícím způsobem. Dotaz je položen autoritativnímu DNS serveru dané domény. Tento dotaz je zaregistrován paketovým procesorem (viz 2.1). Pokud je home-pc do systému DETEC přihlášen tak, jak bylo popsáno v odstavci 3.1, má tento stroj v DNS adresu 0.0.0.1, která se také vrátí

v DNS odpovědi. Paketový procesor zachytí tuto DNS odpověď a před odesláním do aplikační vrstvy tuto odpověď modifikuje. Hodnota 0.0.0.1 informuje paketový procesor o tom, že cílový systém, se kterým se následně bude spojovat, se nachází za překladačem síťových adres.

Paketový procesor si eviduje seznam systémů, se kterými již komunikuje. Musí totiž každý z těchto systémů dokázat odlišit. Jde o to, že pokud z aplikační vrstvy obdrží paket k odeslání, musí na základě informací v paketu obsažených zašifrovat paket správným klíčem a poslat správným směrem. Jedinečným identifikátorem cílového systému je IP adresa. Pro systémy, nacházející se za překladačem síťových adres ale nelze použít libovolnou náhradní IP adresu (mohlo by dojít ke kolizi). Pro tento účel je opět použit adresní prostor 0.0.0.0/8, který garantuje, že nedojde ke kolizi se skutečnými IP adresami. Tímto je zároveň omezen maximální počet cílových systémů za překladačem síťových adres, se kterými komunikuje jediná instalace systému DETEC, na 16,7 milionu.

3.3 Zpracování paketů

Datová struktura pro překlad cílové adresy paketovým procesorem je zobrazena v tab. 3.1.

Tab. 3.1: Tabulka paketového procesoru využívaná pro odchozí a příchozí IP pakety

DNS jméno cílového systému	**IP adresa v aplikační vrstvě**	**IP adresa**	**UDP port**
pc1.detec.firma.cz	0.0.0.84	81.73.58.191	35000
pc2.detec.firma.cz	0.0.0.85	81.73.58.191	35005
public.academy.net	212.46.56.189	212.46.56.189	15888
home-pc.detec.domain.net	0.0.0.86	150.80.135.121	29305

AES klíč	**HMAC klíč**	**Číslo odesílané zprávy**	**Číslo nejvyšší přijaté zprávy**	**Historie přijatých čísel zpráv**
xxxxx	xxxxx	xxxxx	xxxxx	xxxxx
xxxxx	xxxxx	xxxxx	xxxxx	xxxxx
xxxxx	xxxxx	xxxxx	xxxxx	xxxxx
xxxxx	xxxxx	xxxxx	xxxxx	xxxxx

Prvky tabulky využívané pro odchozí IP pakety:

- Klíč: IP adresa v aplikační vrstvě.
- Data: IP adresa, UDP port, AES klíč, HMAC klíč, číslo odesílané zprávy.

Prvky tabulky využívané pro příchozí IP pakety:

- Klíč: IP adresa |[1] UDP port.
- Data: IP adresa v aplikační vrstvě, AES klíč, HMAC klíč, číslo nejvyšší přijaté zprávy, historie přijatých čísel zpráv.

Popis prvků tabulky:

- IP adresa v aplikační vrstvě je IP, kterou obdrží aplikace v DNS odpovědi na svůj dotaz gethostbyname(). V případě, že cílový systém je za NAT, je zde uvedena adresa z prostoru 0.0.0.0/8. V opačném případě je uvedena skutečná cílová IP adresa.
- Cílová IP adresa je skutečná IP adresa ve veřejném adresním prostoru. Jde o veřejnou adresu NAT boxu ukrývajícím cílový systém. V případě, že cílový systém není za NAT, je zde skutečná cílová IP adresa.
- Cílový UDP port je UDP port, na kterém je v cílovém systému otevřen DETEC zabezpečený komunikační tunel. Na tento port jsou zasílány UDP pakety realizující tunel.
- AES klíč je klíčem symetrické šifry AES-128-CTR, kterou je tunelu důvěrnost dat .
- HMAC klíč je klíčem autentizační funkce HMAC-SHA-256, kterou je tunelu zabezpečena autentičnost dat.
- Číslo odesílané zprávy, číslo nejvyšší přijaté zprávy a historie přijatých čísel zpráv jsou parametry, kterými je tunelu zabezpečena ochrana proti útoku zopakováním. Použití těchto čísel je vysvětleno v odstavci 7.1.

Do této datové struktury sahá paketový procesor vždy, když zpracovává odesílaný paket. V případě, že veřejným koncem je NAT, nastaví aplikační vrstva cílovou IP adresu např. na 0.0.0.86. Paketový procesor změní cílovou adresu na 195.178.11.34, data zabezpečí (šifrování, autentizace), zabalí do UDP protokolu a pošle je na port

[1] Operátor zřetězení.

29305. Analogicky, pokud obdrží data se zdrojovou IP adresou 195.178.11.34, protokol UDP zdrojový port 29305, data vybalí z UDP, autentizuje, dešifruje, rekonstruuje původní IP paket, kde změní zdrojovou adresu na 0.0.0.86, přepočítá příslušné kontrolní součty a pošle paket do vyšších vrstev. Podrobně je tento proces popsán v kapitole 7.

4 SPOJOVACÍ SERVER

Spojovací server systému DETEC je jednou z jeho hlavních částí. Plní dvě hlavní funkce:

- Registrace vlastních klientů.
- Zprostředkování spojení k vlastním klientům.

4.1 Registrace vlastních klientů

Registrace vlastních klientů je proces, který podstupuje klient, který požaduje po spojovacím serveru registraci do systému DETEC a získá tak příslušný DNS záznam.

Vlastní klient je klient, který má vazbu na spojovací server. Spojovací server je domovským spojovacím serverem tohoto klienta. To znamená, že pokud se někdo bude chtít spojit s tímto klientem (a tento klient bude za NAT) bude o spojení požádán tento spojovací server.

Cizí klient je klient, který požaduje po spojovacím serveru otevření spojení s jedním z jeho vlastních klientů. DNS jméno spojovacího serveru, ke kterému je konkrétní klient příslušný, lze odvodit z DNS jména klienta. Pokud se klient v DNS objevuje pod názvem pc.detec.domain.net, jeho spojovací server musí mít název connector.detec.domain.net. Klient, který přísluší tomuto serveru, má adresu svého spojovacího serveru v konfiguraci.

Pokud se klient dosud nikdy neregistroval na spojovacím serveru, rozpozná server příslušnost k němu podle spojení s IP adresou brány, která je na spojovacím serveru registrována administrátorem serveru. Spojovací server si uloží MAC adresu síťového rozhraní, ze kterého klient komunikuje a na základě této adresy jej následně vždy identifikuje, ať se připojuje odkudkoliv. Druhou možností je registrace klienta v databázi klientů administrátorem. MAC adresa slouží serveru v podstatě jako primární klíč, podle kterého se vyhledává v databázi klientů. MAC adresu je ale možné uhodnout, a proto je nezbytným autentizačním prvkem použití asymetrické kryptografie. Při první registraci na spojovacím serveru vyžaduje server po klientovi zaslání X.509 certifikátu s veřejným klíčem, kterým bude vždy server šifrovat sekvenci navazování spojení.

4.2 Zprostředkování spojení

Pokud se kdokoliv potřebuje spojit s jedním z vlastních klientů spojovacího serveru, požádá o tuto službu server speciální sekvencí zpráv. Server zprostředkovává toto

spojení jen k vlastním klientům, kteří se nachází ukryti za překladačem síťových adres. Klienti, kteří se nachází ve veřejné síti nepotřebují služby spojovacího serveru a spojení si řídí sami.

4.3 Přidělování DNS jmen

Spojovací server zařizuje přidělení DNS jména pro vlastní klienty, kteří se nachází za překladačem síťových adres v privátní síti. Aby byl takový počítač dosažitelný z veřejné sítě, musí mít přiřazen svůj DNS záznam. Pod uvedeným DNS jménem se s tímto vlastním klientem budou spojovat ostatní. Přidělení DNS jména je řešeno buď staticky, kdy jeden klient obdrží vždy totéž DNS jméno, nebo dynamicky, kdy stejné jméno nemusí být přiděleno. Registrované DNS jméno je klientovi oznámeno při procesu registrace na spojovacím serveru.

4.4 Spojovací služba

Spojovací služba používá k datovým přenosům protokol TCP. Spojení je otevřeno na portu 15551. Spojovací server a klienti provozují tuto službu vždy na tomto portu. Na tento port se připojují jak vlastní klienti, kteří se registrují, tak cizí klienti, kteří zahajují žádost o spojení s vlastním klientem. Před výměnou přihlašovacích zpráv musí klient a server sestavit zabezpečené spojení. Spojovací služba je integrována se všemi koncovými uzly systému DETEC, aby bylo možné provézt finální sestavení zabezpečeného tunelu mezi klienty. Zároveň je použita pro otevření spojení s klienty, kteří se nachází ve veřejné síti. Zde se proces navazování spojení zahajuje právě otevřením spojení a příslušnou sekvencí zpráv na TCP port 15551. Klient, který se nachází ve veřejné síti může používat pro sestavování spojení spojovací server, na kterém je registrován. Jde o případ často migrujících klientů, kteří se občas připojují z veřejné IP adresy. V tomto případě těmto klientům přiděluje spojovací server (pokud je nastaveno statické DNS jméno daného klienta) vždy stejné DNS jméno s uložením IP adresy z prostoru 0.0.0.0/8. Tímto je garantována dostupnost jednoho klienta pod stále stejným DNS jménem ať se fyzicky připojuje k Internetu odkudkoliv.

4.5 Šifrování dat mezi klientem a spojovacím serverem

Klient komunikuje se spojovacím serverem prostřednictvím protokolu TCP, do kterého jsou zabaleny šifrované zprávy protokolu PST. Systém šifrování a řízení šifro-

vání je totožný jako v případě zabezpečeného tunelu Pro zabezpečení komunikace klienta se spojovacím serverem se tedy nevytváří zabezpečený UDP tunel, kde figuruje činnost paketového procesoru, ale veškeré přípravy dat jsou provedeny v aplikační vrstvě a transportovány protokolem TCP.

5 KLIENT

Klient systému DETEC je jednou z jeho hlavních částí. Pracuje prakticky plně transparentně pro uživatele, řídí proces navazování spojení, včetně šifrování jednotlivých zpráv zasílaných mezi systémy a také sestavování zabezpečeného tunelu.

5.1 Registrační služba

Registrační služba je služba, která po spuštění programu provede přihlášení ke svému spojovacímu serveru. Uživatel na tomto místě musí registrační službě zadat své uživatelské heslo, kterým je dešifrován z lokálního úložiště do paměti privátní klíč v PEM formátu pro operace asymetrické kryptografie prováděné v rámci systému DETEC.

Klienti se nemusí vždy přihlašovat ke svému spojovacímu serveru (především klienti na veřejné síti). U těchto klientů se tedy pouze heslem dešifruje privátní klíč a registrace ke spojovacímu serveru se neprovádí. Po připojení klienta na svůj spojovací server dostane klient přiděleno DNS jméno, které může být i statické. Toto je výhodné pro možnost zajištění dostupnosti klienta vždy pod stejným DNS jménem. Této možnosti můžou tedy s výhodou využívat i klienti, kteří často cestují mezi různými sítěmi.

Každý klient vlastní svůj privátní klíč. Klient buď vlastní certifikát podepsaný nějakou důvěryhodnou certifikační autoritou, nebo si může nechat po spuštění programu nechat certifikát vygenerovat. V tomto případě tedy není zajištěna autenticita systému, tedy se neví, zda systém je skutečně tím, za který se vydává (v atributech certifikátu), ale je alespoň zajištěna unikátnost označení klienta systému DETEC.

Klient se všem spojovacím serverům a dalším klientům prokazuje svým certifikátem a svojí MAC adresou, která je[1] globálně jedinečná. Při prvním přihlášení klienta k jeho spojovacímu serveru nebo jinému klientovi, si tento systém přidá do lokální databáze jím zaslaný certifikát. Klíčem je právě MAC adresa klienta. Podle této adresy se následně vyhledává klientův certifikát při dalších připojeních klienta ke spojovacímu serveru nebo jinému klientovi. Pokud nalezený certifikát souhlasí se zaslaným, a je ověřen při navazování spojení digitální podpis dat v atributu s názvem `keyExchangeDone`, tedy náhodné číslo `preMasterSecret` zašifrované pomocí veřejného klíče serveru, je potvrzeno, že se přihlásil skutečně ten, který se pod tímto certifikátem přihlásil do systému poprvé. Nikdo se tedy nemůže vydávat za někoho jiného.

[1] Měla by být.

5.2 Spojovací služba

Klient disponuje vlastní implementací spojovací služby. Spojovací služba slouží klientovi k tomu, aby mohl figurovat v roli serveru v procesu finálního navazování spojení mezi klienty. Pakliže je klient ve veřejné síti (případně za NATem s přesměrovanými porty), může dojít k navázání spojení mezi klienty bez účasti spojovacího serveru. Pro zabezpečení komunikace této klientské verze spojovací služby je použito protokolu PST, 7.1. V tomto případě, kdy je klient ve veřejné síti a je s ním navazováno spojení bez účasti spojovacího serveru, není nutné po výměně kryptografického materiálu (fáze 1, viz 10.2) a sestavení UDP kanálu (fáze 2 a 3 bez procesu UDP hole punching, viz 10.2) realizovat fázi 4 (viz 10.2), neboť kryptografický materiál byl již vyjednán a může být dále používán pro účely zabezpečení UDP tunelu.

5.3 Vysoký stupeň zabezpečení

Klient může akceptovat jen žádosti o spojení s jinými klienty, které zná. Tato znalost je zobrazena ve znalosti veřejných klíčů a MAC adres daných klientů. Pokud by se pokusil s tímto klientem spojit někdo mimo tuto množinu známých, spojení nebude navázáno. Tento fakt vyžaduje konfiguraci, která musí uložit certifikáty okruhu povolených klientů a nastavit tento stupeň zabezpečení. Metoda bezpečného dodání certifikátů není v rámci zájmu této práce. Aby byla tato procedura skutečně bezpečná, měl by celý okruh takto komunikujících klientů disponovat certifikáty, které jsou podepsány důvěryhodnou certifikační autoritou.

6 MECHANISMUS NAVAZOVÁNÍ SPOJENÍ

Pro navázání zabezpečeného spojení mezi dvěma systémy je nutné, aby si tyto systémy mezi sebou, případně ještě se spojovacím serverem, zaslaly několik režijních zpráv. Tyto zprávy jsou zasílány uvnitř následujícího textového protokolu.

6.1 Textový protokol pro zasílání zpráv

Pro výměnu zpráv mezi klienty systémy, a také mezi klienty a spojovacími servery, je použit textově orientovaný protokol, jehož struktura je definována takto:

```
názevAtributu1="hodnota, kde část dat je
na dalším řádku"
dalšíAtribut2="další hodnota"
```

Název atributu je uvozen začátkem řádku a ukončen znakem `=`. Hodnota atributu je uzavřena v uvozovkách. Protokol nepřenáší žádná binární data, tato data jsou kódována do textové podoby kódováním Base64. Uvození dat atributu začíná a končí uvozovkami. Pokud se ve zdrojových datech vyskytuje nový řádek, bude v datech atributu také použit. Jako označení konce celé zprávy je použito znaku `*`.

6.2 Navázání spojení s pomocí spojovacího serveru

Navázání spojení s pomocí spojovacího serveru lze detailněji rozdělit na 4 fáze:

1. Připojení ke spojovacímu serveru a sestavení zabezpečeného spojení se serverem – a) vlastní klient serveru, b) cizí klient serveru.
2. Žádost o spojení s cílovým systémem.
3. Otevření UDP kanálu (UDP hole punching).
4. Zabezpečení UDP kanálu.

Po sestavení spojení protokolem TCP jsou ve fázi 1 vyjednány parametry pro kryptografické zabezpečení datových toků při navazování spojení. Spojovací služba šifruje svá data a balí je do protokolu PST, podle procesu popsaného v kapitole 7.

6.2.1 Fáze 1a: Sestavení spojení a vyjednání klíčů – vlastní klient

Detailní pohled na fázi 1, kde se připojuje ke spojovacího serveru jeho vlastní klient, zobrazuje tab. 6.1.

Zpráva 1.1 je poslána klientem spojovacího serveru po ihned po navázání TCP spojení. Klient v ní uvádí lokální IP adresu síťového rozhraní a dále MAC adresu tohoto rozhraní.

Zpráva přijde serveru a server poté nahlédne do své databáze statických DNS jmen, zda má daný klient uloženo statické jméno. Pokud nemá, vygeneruje dynamické jméno.

Zpráva 1.2 je zaslána serverem klientu. Server v ní uvede výše zmíněné DNS jméno (statické, nebo dynamické).

Klient si po přijmutí této zprávy toto jméno uloží.

Zpráva 1.3 je nejdříve odeslána klientem spojovacímu serveru. Typ zprávy (atribut `messageType`) je identifikován řetězcem `clientHello`. Atribut `keyData` obsa-

Tab. 6.1: Zprávy fáze 1a: Sestavení spojení a vyjednání klíčů – vlastní klient

		`connector.detec.firma.cz`		`pc1.detec.firma.cz` veřejná IP adresa: 81.73.58.191 lokální IP adresa: 10.0.150.135 lokální UDP port: 12540
				Zpráva 1.1: Žádost o připojení k serveru messageType="login" localIp="10.0.150.135" mac="33-33-33-33-33-33"
			←	{TCP, cílový port 15551, nešifrováno}
		Zpráva 1.2: Odpověď na žádost o připojení messageType="loginOk" dnsName="pc1.detec.firma.cz"		
		{TCP, cílový port známý, nešifrováno}	→	
				Zpráva 1.3: Výměna certifikátů messageType="clientHello" keyData="xxxxxxxxxxxxxxxxxxxxxx…" random="xxxxxx…"
			←	{TCP, cílový port 15551, nešifrováno}
		Zpráva 1.3: Výměna certifikátů messageType="serverHello" keyData="xxxxxxxxxxxxxxxxxxxxxx…" random="xxxxxx…"		
		{TCP, cílový port známý, nešifrováno}	→	
				Zpráva 1.4: Výměna klíčů messageType="keyExchange" preMasterSecret="xxxxxxxxxxxxxxxx…"
			←	{TCP, cílový port 15551, šifrováno RSA, podepsáno RSA}
		Výpočet klíčů dle daného algoritmu pro šifry AES a HMAC, provedou obě strany.		
		Zpráva 1.5: Výměna klíčů ukončena messageType="keyExchangeDone" replayProtectionTag="xxxxxx…"		
		{TCP, cílový port známý, šifrováno AES, autentizováno HMAC}	→	
				Zpráva 1.5: Výměna klíčů ukončena messageType="keyExchangeDone" replayProtectionTag="xxxxxx…"
			←	{TCP, cílový port 15551, šifrováno AES, autentizováno HMAC}

huje řetězec certifikátů podle normy X.509v3. Implementace vyžaduje, aby certifikát odesilatele zprávy byl na prvním místě. Každý certifikát následující po předchozím ověřuje předchozí. Dostaneme se tak až na certifikát příslušné certifikační autority, který je podepsán sám sebou. Každý klient systému DETEC disponuje alespoň certifikátem, který si sám vygeneroval při instalaci, který je podepsán sám sebou. Atribut `random` obsahuje náhodně vygenerovanou 32B hodnotu.

Server po přijmutí této zprávy má dvě možnosti. Buď se klient nehlásí poprvé a tak server podle MAC adresy vyhledá dříve uložený certifikát klienta, a uložený certifikační řetězec porovná s poslaným. Provádí se srovnání SHA-256 hashů těchto certikátů. Pokud některý nesedí, spojení bude ukončeno. Pokud se klient přihlašuje poprvé, je jeho certifikát uložen do databáze. Certifikační řetězec je ale nejdříve zkontrolován.

Server poté odešle klientovi tuto zprávu, ve které je typ zprávy pozměněn na řetězec `serverHello`. Atribut `keyData` obsahuje serverův řetězec certifikátů. tribut `random` obsahuje opět náhodně vygenerovanou 32B hodnotu. Tato hodnota musí být nezávislá na hodnotě, kterou zaslal klient serveru.

Klient po přijmutí této zprávy má také dvě možnosti. Buď se již ke svému spojovacímu serveru někdy přihlašoval, a má tak již jeho certifikační řetězec uložený. Poté se provede srovnání stejné jako výše. Pokud se ale klient přihlašuje poprvé, má možnost zkontrolovat certifikát serveru (např. SHA-1 hash certifikátu) se správným certifikátem a poté potvrdí buď přijmutí certifikátu, nebo odmítne. V případě odmítnutí je spojení také ukončeno.

Zpráva 1.4 zahajuje proces výměny šifrovacích klíčů pro symetrické šifrování. Obsahuje materiál, ze kterého si obě strany vypočítají klíče pro symetrickou šifru. Klient vygeneruje náhodnou hodnotu `preMasterSecret`, která je dlouhá 48 B. Tuto hodnotu vloží do zprávy a odešle serveru. Hodnota atributu `preMasterSecret` je šifrována algoritmem RSA veřejným klíčem serveru, který je získán z certifikátu serveru, zaslaného v předchozí zprávě.

Na základě hodnoty `preMasterSecret` a náhodných hodnot z `hello` zpráv provede server i klient výpočet hodnoty `masterSecret`, která je hlavní tajnou hodnotou, ze které se budou odvozovat klíče pro šifrování (AES) a pro autentizaci (HMAC).

```
masterSecret = SHA-256(preMasterSecret, "master secret",
                       Client.random + Server.random)

aesKey = masterSecret[0] ^ masterSecret[31] |
         masterSecret[1] ^ masterSecret[30] | ... | masterSecret[16] ^
         masterSecret[15]

hmacSecret = SHA-256(masterSecret, "hmac secret",
                     Client.random, Server.random)
```

```
hmacKey = hmacSecret[0] ^ hmacSecret[31] |
          hmacSecret[1] ^ hmacSecret[30] | ... | hmacSecret[16] ^
          hmacSecret[15]
```

Symbol ^ reprezentuje operátor XOR. Symbol | reprezentuje zřetězení. Hodnota key je použita jako 128 bitový klíč pro symetrickou šifru AES. Hodnota hmacKey slouží pro výpočty HMAC (klíčem parametrizované ověřovací kódy zpráv, slouží pro ověření integrity zprávy). Bezprostředně po dokončení výpočtu hodnoty `aesKey` a `hmacKey` zašle server klientovi zprávu 1.5.

Zpráva 1.5 (a všechny následující zprávy posílané pomocí TCP protokolu) je šifrována klíčem `aesKey`. Data jsou balena do protokolu PST procesem, který je definován v kapitole 7. Balení dat do protokolu PST řídí spojovací služba v aplikační vrstvě. Na konec zprávy protokolu PST je přidán autentizační kód (HMAC) parametrizovaný klíčem `hmacKey`. Autentizace proběhne nad zašifrovanými daty, aby pro provedení autentizace nebylo nutné nejprve data dešifrovat. Procesor by se totiž musel vždy zatížit dešifrováním, než by bylo možné prohlásit, zda je paket validní. Tato vyšší zátěž vede ke snadnější možnosti DoS útoku.

Jakmile klient obdrží zprávu 1.5 od serveru, autentizuje ji a dešifruje. Pokud vše proběhlo v pořádku a dešifrovaná data odpovídají očekávání, odešle klient serveru tutéž zprávu, kterou sám zašifruje a autentizuje. Server ji obdrží, dešifruje a autentizuje a tím mají oba konce sestaveno zabezpečené spojení.

Důležitým atributem zprávy 1.5 je `replyProtectionTag`. Tento atribut vygeneruje server, a je náhodný. Klient hodnotu tohoto tagu musí zkopírovat do zprávy 1.5, kterou odesílá serveru. Tento tag má velikost 64 bajtů a zajišťuje ochranu proti útoku zopakováním. Útočník takto nemůže serveru odeslat pakety, které dříve zachytil při komunikaci nějakého klienta se serverem. Pokud server obdrží zprávu s nesprávným `replyProtectionTag`, spojení bude ukončeno.

Po ukončení této sekvence zpráv je možné, aby připojený klient mohl kontaktovat jakéhokoli klienta systému DETEC, připojeného ke svému spojovacímu serveru, a aby jakýkoli klient, připojený ke svému spojovacímu serveru, mohl připojeného klienta také kontaktovat. Připojený klient čeká na zprávu o žádost o spojení `connectionRequest` od serveru od jiného klienta, případně sám kontaktuje spojovací server jiného klienta o žádost o spojení k jeho klientovi za překladačem síťových adres.

Tab. 6.2: Zprávy fáze 1b: Sestavení spojení a vyjednání klíčů – cizí klient

home-pc.detec.domain.net veřejná IP adresa: 150.80.135.121 lokální IP adresa: 192.168.2.107 lokální UDP port: 65300		connector.detec.firma.cz
Zpráva 1.1: Výměna certifikátů messageType="clientHello" keyData="xxxxxxxxxxxxxxxxxxxxx…" random="xxxxxx…" {TCP, cílový port 15551, nešifrováno}	→	
		Zpráva 1.1: Výměna certifikátů messageType="serverHello" keyData="xxxxxxxxxxxxxxxxxxxxx…" random="xxxxxx…" {TCP, cílový port známý, nešifrováno}
	←	
Zpráva 1.2: Výměna klíčů messageType="keyExchange" preMasterSecret="xxxxxxxxxxxxxxxx…" {TCP, cílový port 15551, šifrováno RSA, podepsáno RSA}	→	
Výpočet klíčů dle daného algoritmu pro šifry AES a HMAC, provedou obě strany.		
		Zpráva 1.3: Výměna klíčů ukončena messageType="keyExchangeDone" replayProtectionTag="xxxxxx…" {TCP, cílový port známý, šifrováno AES, autentizováno HMAC}
	←	
Zpráva 1.3: Výměna klíčů ukončena messageType="keyExchangeDone" replayProtectionTag="xxxxxx…" {TCP, cílový port 15551, šifrováno AES, autentizováno HMAC}	→	

6.2.2 Fáze 1b: Sestavení spojení a vyjednání klíčů – cizí klient

Detailní pohled na fázi 1, kde se připojuje ke spojovacího serveru cizí klient, zobrazuje tab. 6.2.

Postup připojení je prakticky stejný jako výše, jen klient neposílá cizímu serveru zprávu 1.1 `login` a server neposílá odpověď ve zprávě 1.2.

Zpráva 1.1 (`clientHello`): spojovací server při příjmu této zprávy nezná MAC adresu síťového rozhraní cizího klienta, tedy si ani jeho certifikační řetězec neukládá do databáze. Pouze tento řetězec zkontroluje.

Další zprávy jsou již identické jako u připojení k vlastnímu spojovacímu serveru.

6.2.3 Fáze 2: Žádost o spojení s cílovým systémem

Detailní pohled na fázi 2 zobrazuje tab. 6.3.

V této fázi je spojovací server požádán o spojení s cílovým systémem. Pokud je cílový systém k serveru připojen, bude zahájen proces UDP hole punching, kdy oba klienti otevřou UDP kanál ke spojovacímu serveru.

Zpráva 2.1 (`connectionRequest`). Tato zpráva obsahuje DNS jméno cílového systému v rámci subdomény DETEC. Zpráva také nese informaci o parametrech spojení. Pokud by se totiž oba klienti nacházeli za stejným NAT boxem, nebylo by

Tab. 6.3: Fáze 2: Žádost o spojení s cílovým systémem

home-pc.detec.domain.net veřejná IP adresa: 150.80.135.121 lokální IP adresa: 192.168.2.107 lokální UDP port: 65300		connector.detec.firma.cz lokální UDP port: 36500		pc1.detec.firma.cz veřejná IP adresa: 81.73.58.191 lokální IP adresa: 10.0.150.135 lokální UDP port: 12540
Zpráva 2.1: Žádost o spojení messageType="connectionRequest" destinationName="pc1.detec.firma.cz" client="home-pc.detec.domain.net" localIp="192.168.2.107" localUdp="65300" {TCP, cílový port 15551, šifrováno AES, autentizováno HMAC}	→	**Zpráva 2.1: Žádost o spojení** messageType="connectionRequest" destinationName="pc1.detec.firma.cz" client="home-pc.detec.domain.net" localIp="192.168.2.107" localUdp="65300" udpListeningAt="36500" {TCP, cílový port známý, šifrováno AES, autentizováno HMAC}	→	**Zpráva 2.2: Odpověd na žádost o spojení** messageType="connectionResponse" localIp="10.0.150.135" localUdp="12540" {TCP, cílový port 15551, šifrováno AES, autentizováno HMAC}
	←	**Zpráva 2.2: Odpověd na žádost o spojení** messageType="connectionResponse" localIp="10.0.150.135" localUdp="12540" udpListeningAt="36500" {TCP, cílový port známý, šifrováno AES, autentizováno HMAC}	←	

nutné zasílat data přes veřejný bod NAT boxu, ale mohli by spolu komunikovat přímo. Z tohoto důvodu se při navazování spojení pokoušejí oba klienti navázat spojení v rámci své lokální sítě.

Pokud je iniciátor komunikace ve veřejné síti, obsahují atributy `localIp` a `localUdp` veřejně dostupnou IP adresu resp. otevřený, veřejně dostupný UDP port.

Spojovací server zprávu 2.1 od klienta převezme a doplní ji o číslo svého otevřeného UDP portu. Takto připravenou zprávu přepošle server cílovému systému.

Zpráva 2.2 (`connectionResponse`). Cílový systém tuto zprávu naplní svými lokálními parametry a odešle serveru.

Spojovací server tuto zprávu od cílového systému převezme a doplní ji opět o číslo svého otevřeného UDP portu. Takto připravenou zprávu přepošle server klientu, navazujícím spojení.

Tímto mají obě strany všechny potřebné informace k tomu, aby mohly zahájit odesílání UDP paketů v rámci lokální sítě a také na spojovací server, což zahajuje proces UDP hole punchingu.

6.2.4 Fáze 3: Otevření UDP kanálu (UDP hole punching)

Detailní pohled na fázi 3 zobrazuje tab. 6.4.

Tab. 6.4: Fáze 3: Otevření UDP kanálu (UDP hole punching)

home-pc.detec.domain.net veřejná IP adresa: 150.80.135.121 lokální IP adresa: 192.168.2.107 lokální UDP port: 65300		**connector.detec.firma.cz** lokální UDP port: 36500		**pc1.detec.firma.cz** veřejná IP adresa: 81.73.58.191 lokální IP adresa: 10.0.150.135 lokální UDP port: 12540
Zpráva 3.1: UDP paket messageType="udpPacket" destinationName="pc1.detec.firma.cz" client="home-pc.detec.domain.net" {UDP, cílový port 36500, nešifrováno}	→	Server obdrží přeložený zdrojový port na 20135.		**Zpráva 3.1: UDP paket** messageType="udpPacket" destinationName="home-pc.detec.domain.net" client="pc1.detec.firma.cz"
		Server obdrží přeložený zdrojový port na 11180.	←	{UDP, cílový port 36500, nešifrováno}
	←	**Zpráva 3.2: UDP punch informace** messageType="udpPunchInit" destinationName="pc1.detec.firma.cz" destinationIp="81.73.58.191" destinationPort="11180" {TCP, cílový port známý, šifrováno AES, autentizováno HMAC}		
	→	**Zpráva 3.2: UDP punch informace** messageType="udpPunchInit" destinationName="home-pc.detec.domain.net" destinationIp="150.80.135.121" destinationPort="20135" {TCP, cílový port známý, šifrováno AES, autentizováno HMAC}		
Zpráva 3.3: UDP punch messageType="udpPunch" {UDP na 81.73.58.191:11180, nešifrováno} {UDP na 10.0.150.135:12540, nešifrováno}	→	↔	←	**Zpráva 3.3: UDP punch** messageType="udpPunch" {UDP na 150.80.135.121:20135, nešifrováno} {UDP na 192.168.2.107:65300, nešifrováno}
	←		←	**Zpráva 3.4: UDP tunel sestaven** messageType="udpTunnelEstablished" {UDP na 150.80.135.121:20135, nešifrováno}
Zpráva 3.4: UDP tunel sestaven messageType="udpTunnelEstablished" {UDP na 81.73.58.191:11180, nešifrováno}	→		→	

Fáze 3 je zahájena zasíláním UDP paketů na server a do lokální sítě. Oba koncové body mají otevřen UDP port, oznámený ve zprávě 2.1.

Zpráva 3.1 (`udpPacket`). je oběma stranami zasílána na server. Je poslána protokolem UDP v nezašifrované podobě.Tato zpráva pouze na spojovací server. Pro jistotu je zpráva vyslána vícenásobně, aby bylo možné eliminovat případné náhodné ztracení UDP paketu. UDP paket projde přes překladač síťových adres, který provede změnu zdrojové IP adresy na veřejnou IP a změní také zdrojový UDP port. Pozměněný paket následně dorazí na server, který extrahuje z paketu číslo UDP portu.

Zpráva 3.2 (`udpPunchInit`) předává potřebné informace o veřejných parametrech spojení protistrany. Tuto zprávu zasílá server oběma stranám. S touto zprávou získají obě strany dostatek informací k tomu, aby si mohly navzájem začít zasílat UDP pakety.

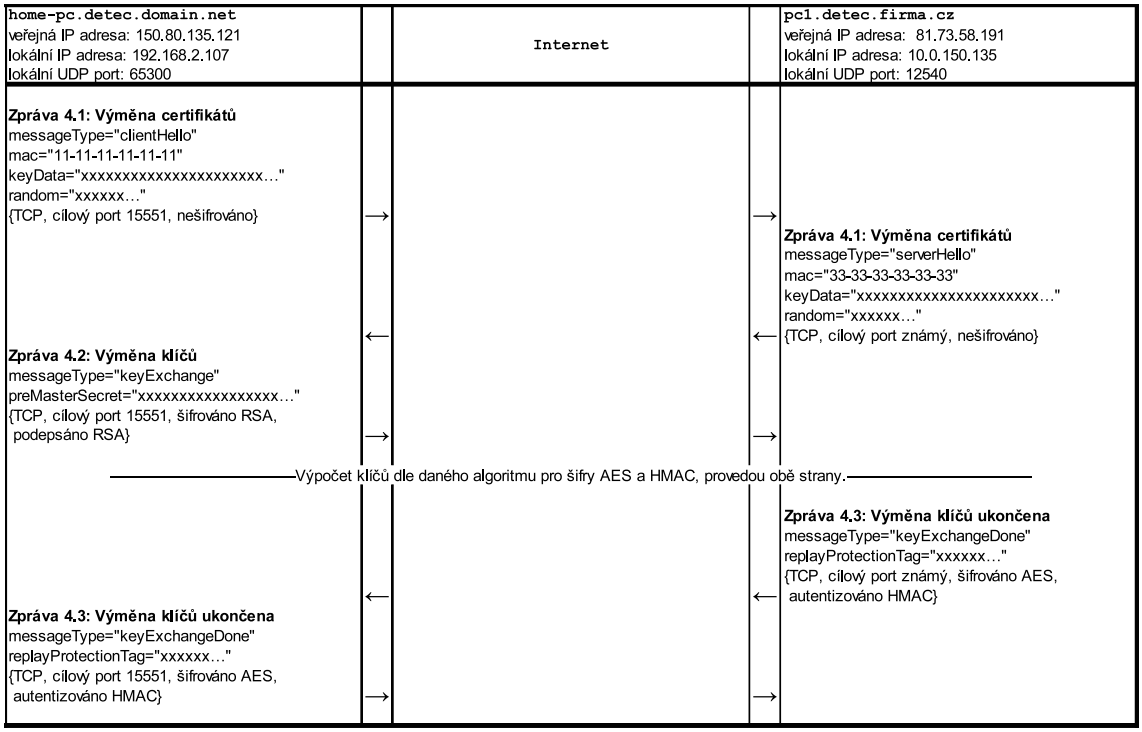
Zpráva 3.3 (`udpPunch`) není šifrována a je zasílána oběma stranami několikanásobně do doby obdržení totožné UDP zprávy od protistrany. UDP paket odeslaný ven z lokální sítě přes překladač síťových adres přinutí překladač vytvořit pravidlo, přes které projdou do lokální sítě pakety protistrany.

Jakmile obdrží klient zprávu 3.3 odešle ihned protistraně **zpráva 3.4** a očekává, že zprávu 3.4 mu odešle také protistrana.

Jakmile zdrojový systém obdrží zprávu 3.4, provádí veškerou komunikaci k cílovému systému prostřednictvím protokolu UDP na daný port. Zprávy jsou zachyceny paketovým procesorem v IP vrstvě, data jsou zabalena do UDP protokolu a odeslána. V tuto chvíli ještě není komunikace nijak šifrována, ale přímý komunikační tunel již existuje.

6.2.5 Fáze 4: Zabezpečení UDP kanálu

Tab. 6.5: Fáze 4: Zabezpečení UDP kanálu



Detailní pohled na fázi 4 zobrazuje tab. 6.5.

Zprávy i proces fáze 4 odpovídá zprávám fáze 1. Jde o zabezpečení komunikačního kanálu, který je realizován prostřednictvím UDP protokolu. Cílový systém v této fázi vystupuje v roli serveru.

Po ukončení této fáze je sestaveno zabezpečené spojení a obě strany mohou kontaktovat libovolné služby spuštěné na libovolných portech a protokolech.

6.3 Navázání spojení bez pomoci spojovacího serveru

Pro navázání zabezpečeného spojení mezi dvěma klienty systému DETEC není ve všech případech nutné žádat o pomoc spojovací server. Pokud se cílový systém nenachází za překladačem síťových adres, není jeho spojovací server vůbec kontaktován. Bez spojovacího serveru je možno navázat v těchto typických případech:

1. Oba klienti na veřejné síti. Navazovat spojení jsou schopni oba klienti.
2. Jeden klient za NATem a druhý na veřejné síti. Navazovat spojení je schopnen pouze klient za NATem.
3. Oba klienti za NATem, z nichž aspoň jeden má správně nastavené přesměrování portů (port forwarding). Navazovat spojení je schopnen pouze klient za NATem bez přesměrování portů, na klienta s přesměrováním.

Jsou to tedy všechny případy, kdy je jeden klient schopný navázat na druhého klienta TCP spojení na daném portu 15551 a kdy je možné druhému klientovi zasílat UDP datagramy na jeho UDP port.

Výše uvedené fáze probíhají u takovéhoto způsobu navazování spojení redukovaně.

Detailní pohled na fázi 1 zobrazuje tab. 6.6 a zredukovanou fázi 2 a 3 zobrazuje tab. 6.7. Fáze 4 sestává pouze z předání klíčů (AES a HMAC), vyjednaných ve fázi 1 pro zabezpečení TCP komunikace mezi klienty, právě pro účely zabezpečení UDP kanálu.

Tab. 6.6: Zprávy fáze 1: Sestavení spojení a vyjednání klíčů mezi klienty

`home-pc.detec.domain.net` veřejná IP adresa: 150.80.135.121 lokální IP adresa: 192.168.2.107 lokální UDP port: 65300		`Internet`		`public.academy.net` veřejná (lokální) IP adresa: 212.46.56.189 veřejný (lokální) UDP port: 15888
Zpráva 1.1: Výměna certifikátů messageType="clientHello" mac="11-11-11-11-11-11" keyData="xxxxxxxxxxxxxxxxxxxxx…" random="xxxxxx…" {TCP, cílový port 15551, nešifrováno}	→		→	
				Zpráva 1.1: Výměna certifikátů messageType="serverHello" mac="22-22-22-22-22-22" keyData="xxxxxxxxxxxxxxxxxxxxx…" random="xxxxxx…" {TCP, cílový port známý, nešifrováno}
	←		←	
Zpráva 1.2: Výměna klíčů messageType="keyExchange" preMasterSecret="xxxxxxxxxxxxxxxx…" {TCP, cílový port 15551, šifrováno RSA, podepsáno RSA}	→		→	
Výpočet klíčů dle daného algoritmu pro šifry AES a HMAC, provedou obě strany.				
				Zpráva 1.3: Výměna klíčů ukončena messageType="keyExchangeDone" replayProtectionTag="xxxxxx…" {TCP, cílový port známý, šifrováno AES, autentizováno HMAC}
	←		←	
Zpráva 1.3: Výměna klíčů ukončena messageType="keyExchangeDone" replayProtectionTag="xxxxxx…" {TCP, cílový port 15551, šifrováno AES, autentizováno HMAC}	→		→	

Tab. 6.7: Zprávy fáze 2, 3: Žádost o spojení a otevření UDP kanálu

`home-pc.detec.domain.net` veřejná IP adresa: 150.80.135.121 lokální IP adresa: 192.168.2.107 lokální UDP port: 65300		`Internet`		`public.academy.net` veřejná (lokální) IP adresa: 212.46.56.189 veřejný (lokální) UDP port: 15888
Zpráva 2.1: Žádost o spojení messageType="connectionRequest" localIp="192.168.2.107" {TCP, cílový port 15551, šifrováno AES, autentizováno HMAC}	→		→	
				Zpráva 2.2: Odpověd na žádost o spojení messageType="connectionResponse" localUdp="15888" {TCP, cílový port známý, šifrováno AES, autentizováno HMAC}
	←		←	
Zpráva 3.1: UDP paket messageType="udpPacket" mac="11-11-11-11-11-11" {UDP na 212.46.56.189:15888, nešifrováno}	→		→	Druhý klient obdrží přeložený zdrojový UDP port na 20230.
				Zpráva 3.2: UDP tunel sestaven messageType="udpTunnelEstablished" {UDP na 150.80.135.121:20230, nešifrováno}
	←		←	
Zpráva 3.2: UDP tunel sestaven messageType="udpTunnelEstablished" {UDP na 212.46.56.189:15888, nešifrováno}	→		→	

7 MECHANISMUS ZABEZPEČENÍ DATOVÝCH TOKŮ

Zabezpečení datových toků je realizováno ve dvou rovinách:

- zabezpečení TCP komunikace spojovací služby,
- zabezpečení UDP tunelu paketovým procesorem.

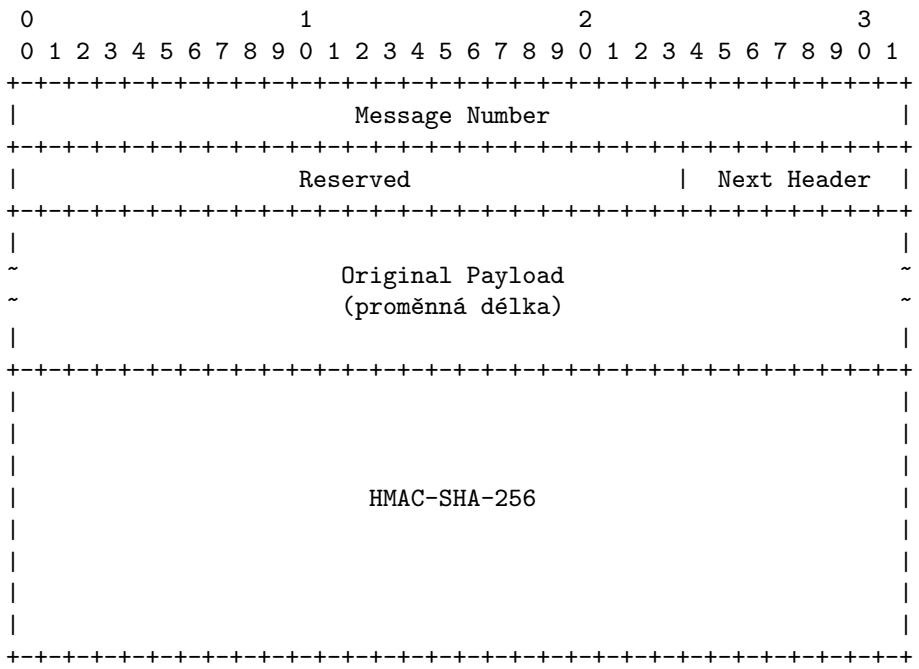
Pro oba tyto případy je využíván stejný způsob zabezpečení. Rozdíl je pouze v tom, že zabezpečení tunelu je řízeno paketovým procesorem z vrstvy síťové, kdy data jsou zasílána tunelem protokolu UDP, naproti tomu zabezpečení komunikace při navazování spojení bez účasti spojovacího serveru nebo spojení se spojovacím serverem je řízeno spojovací službou ve vrstvě aplikační a data jsou odesílána protokolem TCP.

7.1 Protokol pro bezpečné tunely (PST)

Anglický překlad názvu tohoto protokolu je Protocol for Secure Tunnels, zkratka PST. Protokol PST byl navržen pro v rámci disertace [1]. Je navržen tak, aby bylo zaručena důvěrnost a autentičnost dat a aby bylo možné komunikaci chránit proti útoku zopakováním.

Struktura polí protokolu PST je uvedena na obr. 7.1.

Pole *message number* označuje pořadové číslo paketu. Jakmile dosáhne hodnota $2^{32} - 1$, musí být znovu provedena sekvence zpráv navazování zabezpečeného spojení pro přechod z otevřeného kanálu do zabezpečeného, viz. 7.2.2. Tento problém nebyl ve vlastní implementaci vyřešen. Kvůli ochraně proti útoku zopakováním si paketový procesor udržuje krátkou historii těchto čísel. Používá k tomu 32 bitovou hodnotu, kde každý bit je příznakem, zda zpráva daného pořadového čísla již byla obdržena (příznak má hodnotu 1) či nikoliv (příznak obsahuje hodnotu 0). Vyšší bity obsahují příznak o obdržení starších zpráv. Jakmile je obdržena další zpráva, bity jsou rotovány doleva o takový počet bitů, jaký je rozdíl mezi největším sekvenčním číslem dříve obdržené zprávy a právě obdržené zprávy. Paketový procesor např. obdržel zprávy v pořadí 1, 2, 3, 8, 5, 29, 27 (data jsou zasílána UDP protokolem, a tedy mohou být doručena v jiném pořadí, než byla odeslána, některé paketu se mohou na cestě ztratit úplně). Data příznaku jsou tedy nejprve 3x rotována o jeden bit vlevo, pak o 5 bitů, zpráva 5 jen nastaví příslušný příznak (rotování se neprovádí). Zpráva 29 způsobí rotování o 21 bitů (protože nejvyšší číslo před ní bylo 8), zpráva 27 opět jen nastaví příslušný příznak, atd. Proces je ukázán na obr. 7.2.



Obr. 7.1: Struktura dat protokolu PST [1]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Číslo bitu / Obdržená zpráva	Nejvyšší dosud obdržená zpráva
																															1	1	1
																														1	1	2	2
																													1	1	1	3	3
																								1	1	1	0	0	0	0	1	8	8
																								1	1	1	0	1	0	0	1	5	8
			1	1	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	29	29
			1	1	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	27	29

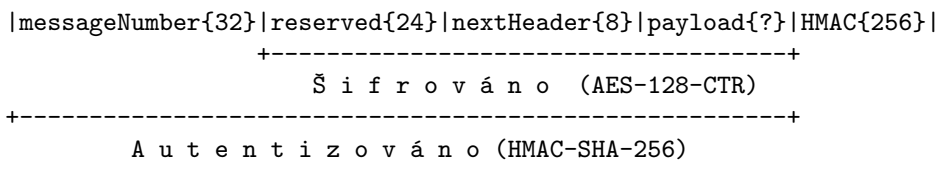
Obr. 7.2: Princip uchovávání historie obdržených zpráv [1]

Pole *next header* reprezentuje číslo protokolu[1], který je zabalen v protokolu PST.

Pole *HMAC-SHA-256* obsahuje hodnotu HMAC vypočtenou za použití algoritmu SHA-256. Výpočet je proveden podle [8].

Na obr. 7.3 jsou zobrazeny pole protokolu PST, která jsou šifrována a pole která jsou autentizována. Byla vybrána metoda nejdříve data zašifrovat a poté až provést autentizační proces. Kdyby totiž případný útočník zasílal velké množství paketů, pak by musel klient, než by zjistil, že obdržel falešný nebo poškozený paket, nejprve paket dešifrovat a teprve potom autentizovat, což je náročnější na procesorový výkon. Systém by tak byl méně odolný vůči útokům typu odepření služby (Denial of Service – DoS).

[1] číslování dle IANA



Obr. 7.3: Šifrovaná vs. autentizovaná pole protokolu PST (čísla uvádí počet bitů) [1]

7.1.1 Výpočty kontrolních součtů IP, UDP, TCP

Při balení dat do protokolu PST a naopak je nutné přepočítávat příslušné kontrolní součty použitých síťových protokolů. Pro protokol IP vždy, a dále, dle neseného protokolu, buď pro protokol UDP, nebo TCP.

7.1.2 Nastavení MTU

Celková režie protokolu PST je za použití autentizační funkce HMAC-SHA-256 320 bitů. K té je nutné přičíst režii protokolu UDP, do kterého je protokol PST zabalen, která činí 64 bitů. Celkem je to tedy 384 bitů (48 B).

Přechodu na bezpečnější verze (případně jiné algoritmy než AES) šifrovacího algoritmu AES (např. AES-256-CTR) nic nebrání, neboť v případě použití módu CTR není potřeba nijak snižovat MTU. Při použití systému i na nejlevnějších počítačích, prodávaných v nynější době, nehrozí žádné citelné snížení přenosové rychlosti.

Při přechodu na bezpečnější verzi autentizační funkce by již bylo potřeba odpovídajícím způsobem MTU dále snižovat. V případě HMAC-SHA-512 by to znamenalo snížit MTU o 640 bitů (80 B).

7.1.3 Fragmentace IP paketů

Pro Ethernet má MTU hodnotu 1500, to znamená, že maximální délka dat v Ethernetovém rámci odpovídá 1500 B. Maximální délka Ethernetového rámce je tato hodnota plus Ethernetová hlavička, která je rovna 14 B. Aby nebylo nutné pakety fragmentovat, musí být MTU snížena o požadovanou hodnotu. V našem případě o 48 B, na 1452 B. Odchozí pakety nám tedy odchází nefragmentované, ale příchozí mohou přijít fragmentované. Zpětné složení lze provést např. dle algoritmu [7].

Problém, který by mohl vzniknout případnou fragmentací IP paketů, zasílaných systémem DETEC, nebyl v rámci vlastní implementace systému řešen.

7.2 Paketový procesor

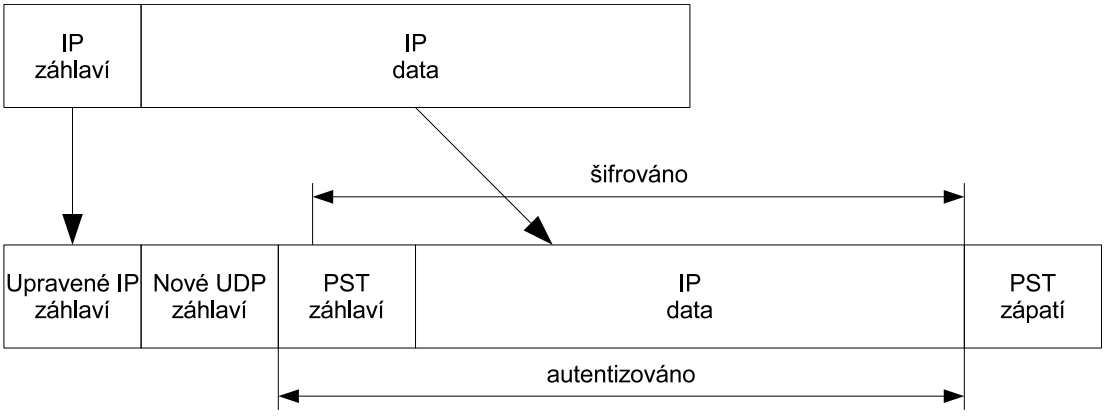
Paketový procesor je spuštěn v okamžiku startu systému. Zachytává odchozí pakety, řídí požadavky na navazování spojení s koncovým systémem, kdy o navázání spojení žádá spojovací službu. Pracuje ve dvou režimech.

7.2.1 Režim Otevřený UDP kanál

V tomto režimu *paketový procesor* pouze vezme pakety a balí je do protokolu UDP a odesílá cílovému systému. V režimu otevřeného kanálu tedy nejsou zprávy nijak šifrovány ani autentizovány, ale existuje již kanál, kterým prochází komunikace všech vyšších protokolů, které využívají protokol IP, fyzicky přes jediný UDP port. Na obdržené pakety na patřičném UDP portu reaguje tak, že je rozbalí z protokolu UDP, rekonstruuje původní paket a předá jej vyšším vrstvám.

7.2.2 Režim Zabezpečený UDP kanál

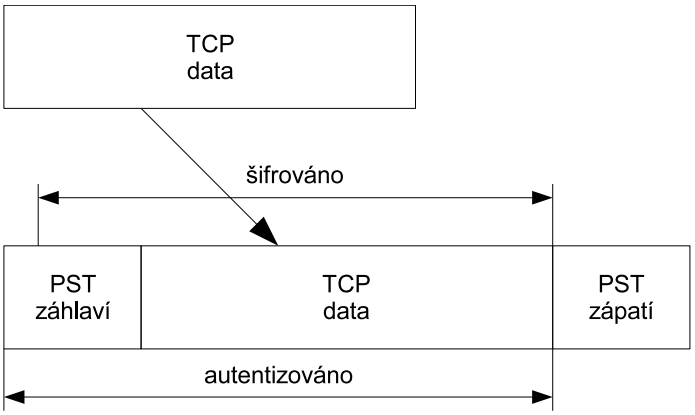
V tomto režimu *paketový procesor* šifruje a autentizuje odchozí pakety a dešifruje a ověřuje autenticitu příchozích paketů, opět přes jediný UDP port. Zabezpečený kanál je sestaven pomocí *spojovací služby* klienta.



Obr. 7.4: Princip zabezpečení režimu zabezpečený UDP kanál

7.3 Spojovací služba

Spojovací služba nepoužívá UDP tunel, ale přímé spojení protokolem TCP. Po vyjednání kryptografických parametrů ve fázi 1, viz. 7.2.2, jsou zprávy baleny do protokolu PST, viz. 7.1 a odesílány. Šifrování a autentizace dat probíhá pořád stejně.



Obr. 7.5: Princip zabezpečení spojovací služby

7.4 Proces šifrování paketů

Pakety jsou šifrovány algoritmem AES-128 v proudovém režimu CTR.

Celý proces vyžaduje výpočet proudu klíčů, který je generován za pomocí pole *message number* protokolu PST a tajného klíče.

Proud klíčů k je generován podle vztahu:

$$k_j = E(K, j||i||s); j = 0..l/128, \tag{7.1}$$

kde j je číslo (8 b) aktuálního 128 b bloku dat, která mají být zašifrována, i odpovídá hodnotě *message number* (32 b), s je konstanta, kterou tvoří prvních 88 b šifrovacího klíče, l je počet bitů dat na zašifrování. Tyto hodnoty jsou zřetězeny do 128 bitů dlouhé sekvence bitů, která je vstupním blokem pro šifru AES. Na základě šifrovacího klíče je tento vstupní blok zašifrován (parametrizace je provedena číslem zprávy) a výstupem je tzv. segment proudu klíčů (k_j), jehož délka je 128 bitů. Proud klíčů je generován na obou komunikujících stranách stejně, a proto jsou obě schopny správě šifrovat i dešifrovat data. Tato finální operace (šifrování/dešifrování) je provedena podle rovnic

$$c_j = p_j \bigoplus k_j, \tag{7.2}$$

$$p_j = c_j \bigoplus k_j, \tag{7.3}$$

, kde c_j je j-tý zašifrovaný blok, p_j je j-tý dešifrovaný blok.

8 EXPERIMENTY, MĚŘENÍ

8.1 Navázání a provoz zabezpečeného připojení

V rámci ověření vytvořené implementace systému DETEC, tedy klienta systému, spojovacího serveru a vazby tohoto systému na systém DNS, byly provedeny testy korektního navazování spojení vedoucí k sestavení zabezpečeného tunelu mezi klienty systému.

Tunelem systému bylo po navázání spojení zkoušeno přenášet několik síťových služeb (protokolů). Jmenovité ICMP pakety *echo request* a *echo response*, FTP, HTTP a „Sdílení souborů a tiskáren“ na MS Windows XP využívající protokol SMB (Server Message Block). Byly tedy mezi systémy posílány zprávy přikazu PING, a systémy si přistupovaly navzájem na HTTP a FTP servery na nich běžící a využívaly služeb sdílení souborů na Windows.

8.1.1 Testovací virtuální síť

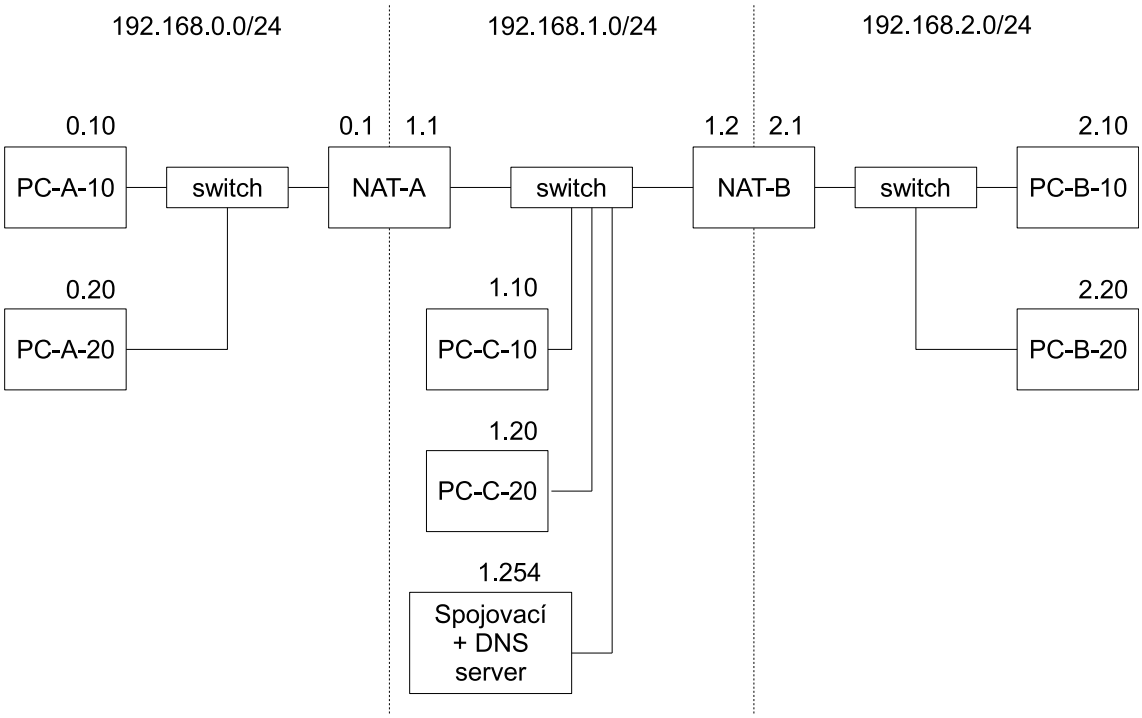
V rámci testování systému byla vytvořena jednoduchá testovací síť, v programu Microsoft Virtual PC 2007 [20]. Všichni klienti náleželi ke spojovacímu serveru s IP adresou 192.168.1.254 a DNS jménem connector.detec.domain.test.

Služba překladu adres byla řešena softwarově programem NAT32 [14], nebo pomocí „Sdílení připojení k internetu“ (ICS – Internet Connection Sharing) na OS Windows XP. Experimentálně bylo zjištěno, že obě řešení podpororují techniku UDP hole punching, ale ne již techniku hairpin překladu.

8.1.2 Cílový klient systému na veřejné síti

V této variantě může být zdrojový systém volitelně připojen ke svému spojovacímu serveru. Cílový systém také ke svému serveru může, ale nemusí být připojen. Výhodou připojení cílového klienta ke svému spojovacímu serveru lze spatřovat v tom, že po připojení může dostat klient od svého spojovacího serveru přiděleno statické DNS jméno, a lze na něj pak tedy přistupovat pomocí tohoto DNS jména, ať už se momentálně připojuje z jakékoli sítě.

Korektně navázané spojení a výše zmíněné služby byly ověřeny mezi dvěma počítači, které byly přímo propojeny propojeny kříženým UTP kabelem, poté na výše zmíněné virtuální síti a také po Internetu mezi dvěma počítači za NATem (zde neprobíhalo ověření sdílení na Windows; mezi PC 16 směrovačů), kde cílový PC měl přesměrované porty využívané systémem DETEC (v tomto případě TCP 15551 a UDP 50000).



Obr. 8.1: Jednoduchá virtuální testovací síť

Na virtuální síti byly otestovány všechny možnosti takovéhoto spojení, bez připojení klientů k serveru i s připojením. Byly zkoušeny všechny služby (protokoly) zmíněné výše.

Všechny tyto testy proběhly správně, a tak po navázání spojení byl sestaven zabezpečený komunikační tunel mezi patřičnými klienty.

8.1.3 Cílový klient systému za překladačem síťových adres

Tato varianta byla testována pouze na virtuální síti. Pro tuto možnost musí mít oba klienti navázáno spojení se svým spojovacím serverem.

V této možnosti tedy přistupovali klienti ze sítě 192.168.1.0/24 na klienty na sítích 192.168.0.0/24 a 192.168.2.0/24, dále navzájem na sebe počítače z těchto dvou sítí za NATy, případně ještě jeden klient na druhého klienta za stejným NATem.

Všechny tyto testy proběhly také správně a po navázání spojení byl sestaven zabezpečený komunikační tunel mezi patřičnými klienty. Opět byly zkoušeny všechny služby (protokoly) zmíněné výše.

8.2 Transportní výkon

8.2.1 Testovací podmínky

Pro otestování výkonu přenosového tunelu systému DETEC jsem tento systém srovnal s IPsec implementací ve Windows XP a s SSL/TLS implementací v OpenVPN [16].

Přenosová rychlost jednotlivých systémů řešících zabezpečení byla testována pomocí programu Iperf [13].

V rámci semestrálního projektu byl otestován výkon na jednom PC, pomocí technologie virtualizace v programu VMware Server [21]. Na VMware Serveru běžely dva PC, které sdílely hardwarové prostředky hlavního PC. Na klientu programu Iperf byl spuštěn omezovač šířky pásma (traffic shaper) SoftPerfect Bandwidth Manager [17], pomocí kterého byla nastavována maximální šířka pásma sítě. Výsledky tohoto měření byly převzaty. Tato konfigurace je označena jako 2. konfigurace a je to prakticky simulace provozování těchto systémů na nepříliš výkonných počítačích.

V rámci konfigurace označené jako 1. konfigurace byl otestován výkon na dnešní době běžných počítačích, mezi stolním PC a notebookem, propojenými přímo kříženým UTP kabelem. Obě použité konfigurace jsou vidět dále.

1. konfigurace (2 PC):

- PC 1 – stolní PC: AMD Athlon 64 X2 3800+ (2 GHz), 2 GB DDR2 667 MHz, Windows XP Professional SP2.
- PC 2 – notebook: Intel Pentium Dual-Core T2130 (1,86 GHZ), 1 GB DDR2 667 MHz, Windows XP Professional SP2.

2. konfigurace (1 PC)[1]:

- PC: AMD Athlon64 X2 3800+, 2 GB DDR2 667 MHz, Windows XP Professional SP2,
- VMware (2 PC): 512 MB RAM, Windows XP Professional SP2.

Nastavení DETEC:

- 150 PC zadaných ve struktuře pro vyhledávání[2],
- Šifrovací/autentizační funkce – AES-128-CTR/HMAC-SHA-256.

Nastavení IPsec:

[1] Převzato ze semestrálního projektu.
[2] Pouze v konfiguraci 2.

- ESP protokol, transportní režim,
- šifrovací/autentizační funkce – 3DES-CBC/HMAC-SHA-1.

Nastavení OpenVPN:

- UDP protokol, TAP zařízení, LZO komprese,
- šifrovací/autentizační funkce – AES-128-CBC/HMAC-SHA-1.

Nastavení Iperf:

- server: `iperf -s -w 65535`,
- klient: `iperf -c server -t 30 -w 65535 -f kb`.
- Protokol TCP, velikost okna 65535 B, čas měření 30 s.

8.2.2 Širší přenosová pásma

Výsledky pro širší přenosová pásma pro obě konfigurace jsou zobrazeny v tab. 8.1, 8.2 a na obr. 8.2, 8.3. Jak si můžeme povšimnout, limit tunelu DETEC systému byl na 1. konfiguraci 91 Mb/s a na 2. konfiguraci 37 Mb/s. Je zde tedy možno vidět, že dosažená přenosová rychlost je značně závislá na výkonu počítače (především CPU), a to u všech systémů řešících zabezpečení datových toků. Na běžných počítačích tedy nelze na typické šířce pásma 100 Mb/s pozorovat citelný pokles přenosové rychlosti oproti nešifrovanému kanálu. Bylo dosaženo vyšší rychlosti než u řešení IPsec a OpenVPN, nejspíše i z důvodu menší režie protokolu PST. Předpokládám, že je to ale především díky použití velmi efektivního a rychlého prostředí WinpkFilter [18], tedy zachytávače paketů.

8.2.3 Užší přenosová pásma

Výsledky pro širší přenosová pásma pro obě konfigurace jsou zobrazeny v tab. 8.3, 8.4 a na obr. 8.4, 8.5. Zde již jsou výsledky prakticky

8.2.4 Míra negativního dopadu zabezpečení

Na obr. 8.6, 8.7 můžeme pro obě použité konfigurace vidět, o kolik % je dosažená přenosová rychlost jednotlivých systémů nižší než v případě nešifrovaného kanálu

Tab. 8.1: Dosažené přenosové rychlosti na širších přenosových pásmech – 1. konfigurace

Šířka pásma (Mb/s)	Kanál zabezpečovacího systému			
	Nešifrováno	**DETEC**	IPsec	OpenVPN
	Přenosová rychlost (kb/s)			
100	94600	91185	46869	41729
70	69567	69473	46912	40373
50	49696	49677	47048	41268
30	29832	29810	29100	26181
10	9941	9939	9710	8716

Tab. 8.2: Dosažené přenosové rychlosti na širších přenosových pásmech – 2. konfigurace

Šířka pásma (Mb/s)	Kanál zabezpečovacího systému			
	Nešifrováno	**DETEC**	IPsec	OpenVPN
	Přenosová rychlost (kb/s)			
100	95137	37941	33945	21290
50	49609	37589	33928	20524
30	29794	29742	29039	21072
10	9939	9936	9712	8657
5	4969	4967	4853	4338

8.3 Analýza paketů zasílaných tunelem systému

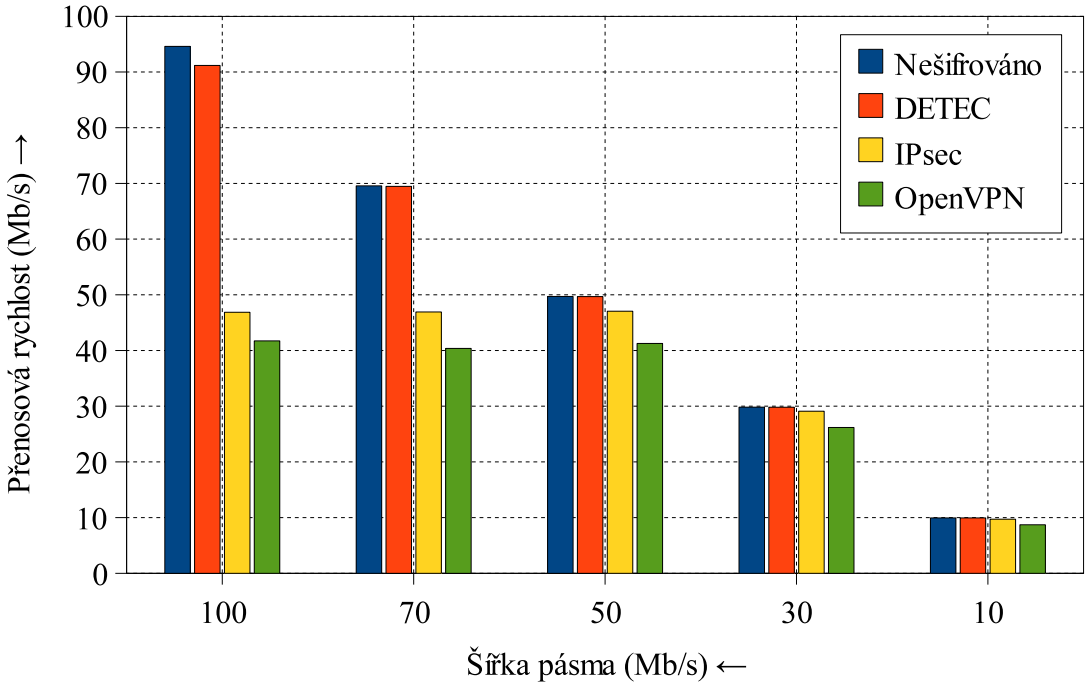
Z důvodu prokázání, že komunikační tunel systému DETEC je bezpečný, bylo provedeno zachycení několika nešifrovaných i šifrovaných rámců programem Wireshark [19]. Je zde možno vidět u protokolu ICMP jeho zprávu typu `echo request`. Na obr. 8.8 můžeme vidět dva nešifrované rámce zachycené v jiném čase. Jak si můžeme povšimnout, datová část ICMP paketu je stejná. 32 B těchto dat je vidět od offsetu 0x2A. Na dalším obr. 8.9 můžeme naopak vidět rámce šifrované. Zde je také vidět, že šifrovaná data dvou „stejných“ rámců lišících se v čase, jsou úplně jiná. Rámec je o 0x30 B, tedy o 48 B delší než nešifrovaná verze. To je přesně součet režie protokolu UDP (8 B), záhlaví PST (16 B) a záhlaví PST (32 B), tedy celkové režie použité v přenosovém kanálu paketového procesoru mezi dvěma klienty systému DETEC.

Tab. 8.3: Dosažené přenosové rychlosti na užších přenosových pásmech – 1. konfigurace

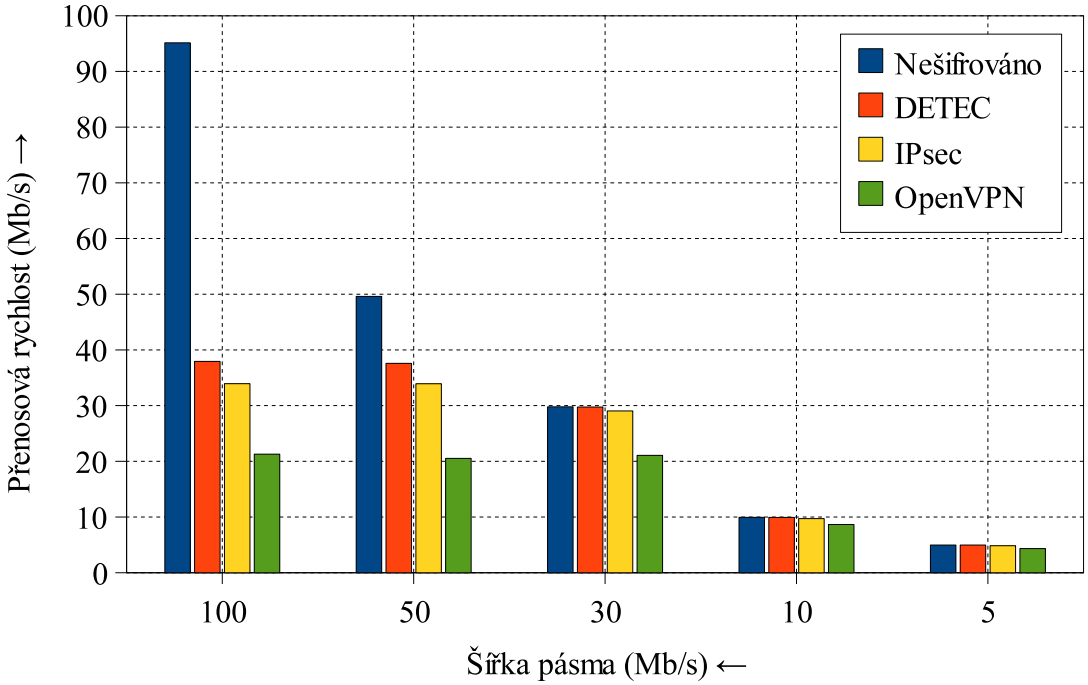
Šířka pásma (kb/s)	Kanál zabezpečovacího systému			
	Nešifrováno	**DETEC**	IPsec	OpenVPN
	Přenosová rychlost (kb/s)			
5000	4969	4975	4856	4364
2000	1989	1989	1943	1757
1000	995	994	971	878
512	509	509	497	449
256	254	254	248	224

Tab. 8.4: Dosažené přenosové rychlosti na užších přenosových pásmech – 2. konfigurace

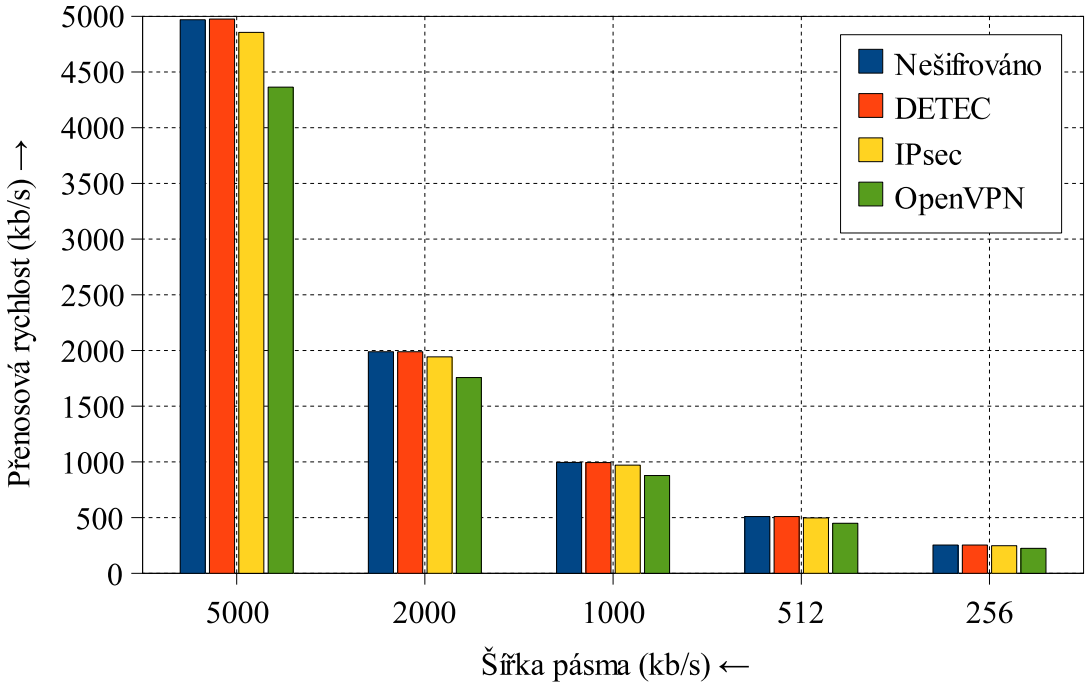
Šířka pásma (kb/s)	Kanál zabezpečovacího systému			
	Nešifrováno	**DETEC**	IPsec	OpenVPN
	Přenosová rychlost (kb/s)			
2000	1987	1987	1942	1736
1000	994	994	971	869
512	509	509	497	445
256	254	254	246	224
64	63	63	62	56



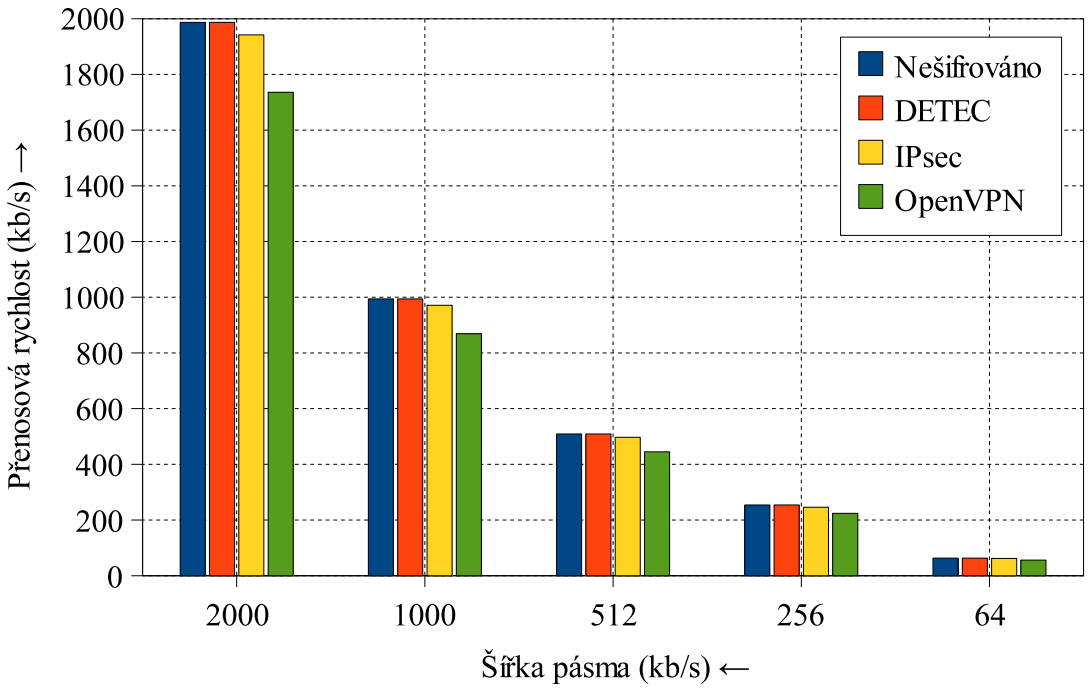
Obr. 8.2: Dosažené přenosové rychlosti na širších přenosových pásmech – 1. konfigurace



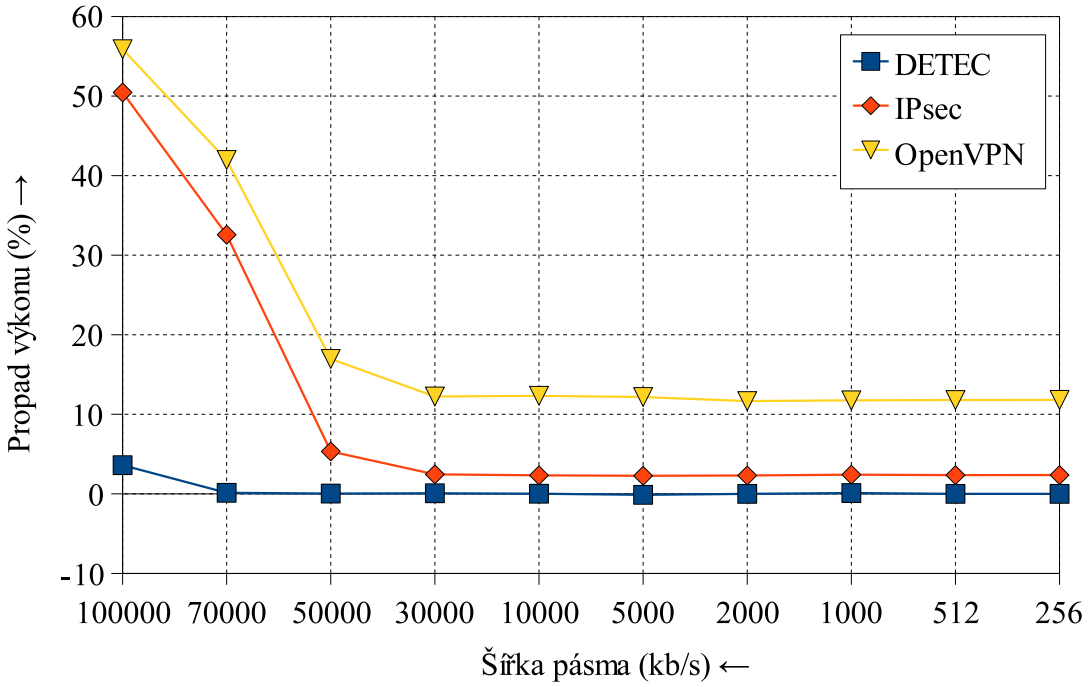
Obr. 8.3: Dosažené přenosové rychlosti na širších přenosových pásmech – 2. konfigurace



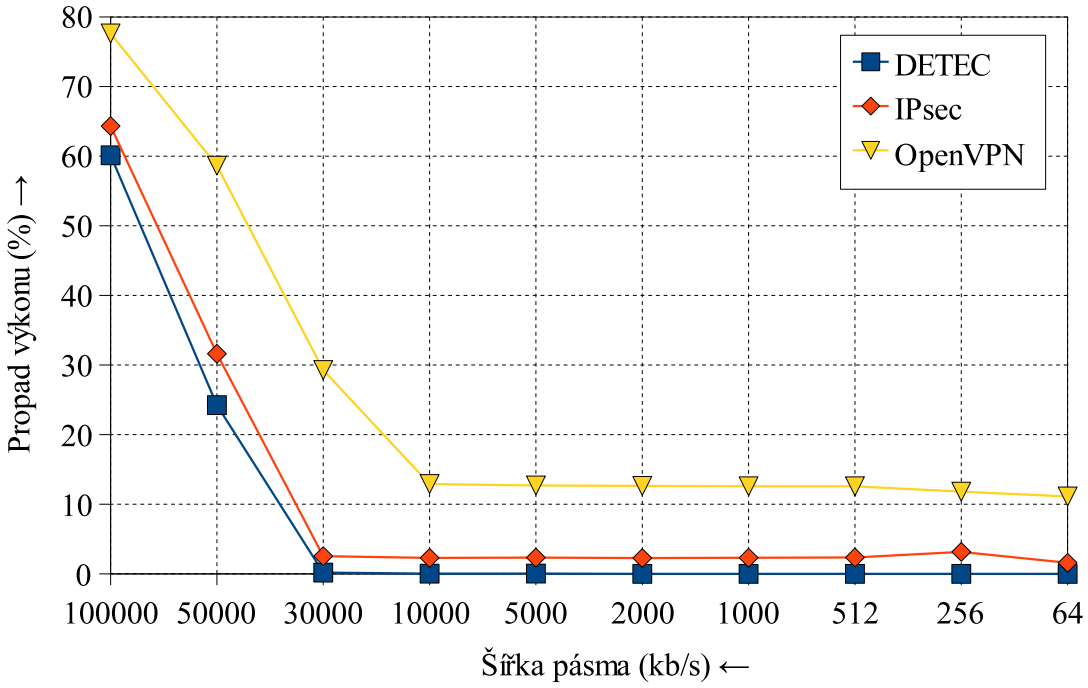
Obr. 8.4: Dosažené přenosové rychlosti na užších přenosových pásmech – 1. konfigurace



Obr. 8.5: Dosažené přenosové rychlosti na užších přenosových pásmech – 2. konfigurace



Obr. 8.6: Míry negativního dopadu zabezpečení na výkon – 1. konfigurace



Obr. 8.7: Míry negativního dopadu zabezpečení na výkon – 2. konfigurace

```
0000  00 03 ff 7a 91 f6 00 03  ff 2a a6 54 08 00 45 00   ...z.... .*.T..E.
0010  00 3c 00 d2 00 00 80 01  df cb ac 10 01 01 ac 10   .<...... ........
0020  01 02 08 00 3b 5c 03 00  0f 00 61 62 63 64 65 66   ....;\.. ..abcdef
0030  67 68 69 6a 6b 6c 6d 6e  6f 70 71 72 73 74 75 76   ghijklmn opqrstuv
0040  77 61 62 63 64 65 66 67  68 69                    wabcdefg hi

0000  00 03 ff 7a 91 f6 00 03  ff 2a a6 54 08 00 45 00   ...z.... .*.T..E.
0010  00 3c 00 d3 00 00 80 01  df ca ac 10 01 01 ac 10   .<...... ........
0020  01 02 08 00 3a 5c 03 00  10 00 61 62 63 64 65 66   ....:\.. ..abcdef
0030  67 68 69 6a 6b 6c 6d 6e  6f 70 71 72 73 74 75 76   ghijklmn opqrstuv
0040  77 61 62 63 64 65 66 67  68 69                    wabcdefg hi
```

Obr. 8.8: Dva zachycené Ethernet rámce ICMP protokolu (typ echo request), zachycené v jiném čase – nešifrovaný kanál

```
0000  00 03 ff 7a 91 f6 00 03  ff 2a a6 54 08 00 45 00   ...z.... .*.T..E.
0010  00 6c 00 d4 00 00 80 11  df 89 ac 10 01 01 ac 10   .l...... ........
0020  01 02 f0 6a f0 6a 00 58  00 00 01 00 00 00 25 05   ...j.j.X ......%.
0030  4d c6 6b 80 d0 3e 32 21  e9 86 c5 2a 4c b9 57 6e   M.k..>2! ...*L.Wn
0040  76 f6 38 61 87 07 58 f8  96 e9 71 0e 15 42 81 ce   v.8a..X. ..q..B..
0050  4d d3 dc 95 1f 96 7d 37  a9 19 cc 8d 6a c0 e3 52   M.....}7 ....j..R
0060  74 ad 8d 47 d9 7e 23 f1  de 91 c2 51 e8 47 c5 7f   t..G.~#. ...Q.G..
0070  30 37 52 eb 55 a4 3c 35  5e a9                    07R.U.<5 ^.

0000  00 03 ff 7a 91 f6 00 03  ff 2a a6 54 08 00 45 00   ...z.... .*.T..E.
0010  00 6c 00 d5 00 00 80 11  df 88 ac 10 01 01 ac 10   .l...... ........
0020  01 02 f0 6a f0 6a 00 58  00 00 02 00 00 00 a8 5b   ...j.j.X .......[
0030  13 66 51 f7 53 07 94 5d  4a d9 8f 22 62 d8 f8 3a   .fQ.S..] J.."b..:
0040  9d 4b d0 c6 3e ad 45 34  19 f6 5e 38 f8 55 b0 14   .K..>.E4 ..^8.U..
0050  04 c3 1e 17 0a a6 de 37  94 fb 74 75 22 53 d3 8b   .......7 ..tu"S..
0060  c4 e6 54 f0 0b e2 1c 64  40 68 4d 93 88 fe af d7   ..T....d @hM.....
0070  90 bf 5d 8e 24 72 cd 44  fa 25                    ..].$r.D .%
```

Obr. 8.9: Dva zachycené Ethernet rámce ICMP protokolu (typ echo request), zachycené v jiném čase – šifrovaný DETEC kanál

9 ZÁVĚR

Cílem práce bylo implementovat obecný zabezpečený peer-to-peer komunikační systém. Tento systém byl navržený v rámci disertační práce [1].

Systém kryptograficky chrání síťové toky aplikací, které o tomto zabezpečovacím mechanismu vůbec nevědí. Pro jednotlivé aplikace je toto vše transparentní.

Navázání spojení mezi dvěma klienty systému probíhá prakticky zcela automatizovaně. Uživatel jen spustí systém a volitelně se připojí ke svému spojovacímu serveru. Poté již může spustit požadovanou síťovou aplikaci, kde zadá buď IP adresu, nebo DNS jméno cílového systému. Systém DETEC se již v případě, že na druhé straně také běží klient tohoto systému, postará o korektní navázání spojení a sestavení zabezpečeného tunelu. Uživatel na tomto místě pouze po spuštění programu klienta zadá heslo ke svému privátnímu klíči a kontroluje (schvaluje, či odmítá) certifikáty uživatelů, se kterými chce komunikovat a kteří chtějí komunikovat s ním. Tato kontrola probíhá ale pouze jednou (poprvé) u každého klienta, certifikát klienta (serveru) se poté uloží do lokální databáze a následující žádost o spojení již probíhá zcela automatizovaně a uživatel není vůbec obtěžován.

Obecné principy systémy jsou popsány v kapitolách 2 a 3. Popisu klienta se věnuje více kapitola 5 a popisu spojovacího serveru kapitola 4. Mechanismus navazování spojení rozebírá kapitola 6 a mechanismus zabezpečení datových toků kapitola 7. Testování korektního navazování spojení a sestavení komunikačního tunelu, při různých síťových umístění klientů, je ukázáno v kapitole 8. V této kapitole jsou také provedeny testy rychlosti tunelu systému a srovnány s rychlostmi dosaženými systémy IPsec a OpenVPN při podobném zabezpečení komunikačního tunelu.

Lze konstatovat, že cíl této diplomové práce, a to implementace zabezpečeného komunikačního systému v jazyce C++ na OS Windows XP, byl splněn. Řešení sice není (především z časových důvodů) dotaženo do úplného konce a jistě není také optimální, ale dovolím si tvrdit, že většina problému byla vyřešena. Především kód serveru a tedy i spolupráce klient–server není dotažena do úplného konce. V rámci testování je ale vytvořená implementace použitelná.

LITERATURA

Literatura:

[1] Cvrk, L. *Kryptografické metody zabezpečení datových přenosů.* Disertační práce, 2006.

[2] Ford, B., Srisuresh, P., Kegel, D. *Peer-to-Peer Communication Across Network Address Translators* [online]. Publikováno 2005-02-17 [cit. 2007-12-20]. Dostupné z URL: <http://www.bford.info/pub/net/p2pnat/>.

[3] Pužmanová, R. *Bezpečnost ve VPN: IPSec versus SSL* [online]. Publikováno 2006-09-12 [cit. 2007-12-20]. Dostupné z URL: <http://www.dsl.cz/clanky-dsl/clanek-515/Bezpecnost-ve-VPN:-IPSec-versus-SSL>.

[4] RFC 768. *User Datagram Protocol* [online]. Publikováno 1980-08-28. Dostupné z URL: <http://www.faqs.org/rfcs/rfc768.html>.

[5] RFC 791. *Internet Protocol* [online]. Publikováno 1981-09. Dostupné z URL: <http://www.faqs.org/rfcs/rfc791.html>.

[6] RFC 793. *Transmission Control Protocol* [online]. Publikováno 1981-09. Dostupné z URL: <http://www.faqs.org/rfcs/rfc793.html>.

[7] RFC 815. *IP datagram reassembly algorithms* [online]. Publikováno 1982-07. Dostupné z URL: <http://www.faqs.org/rfcs/rfc815.html>.

[8] RFC 2104. *HMAC: Keyed-Hashing for Message Authentication* [online]. Publikováno 1997-02. Dostupné z URL: <http://www.faqs.org/rfcs/rfc2104.html>.

[9] RFC 2246. *The TLS Protocol Version 1.0* [online]. Publikováno 1999-01. Dostupné z URL: <http://www.faqs.org/rfcs/rfc2246.html>.

[10] RFC 2401. *Security Architecture for the Internet Protocol* [online]. Publikováno 1998-11. Dostupné z URL: <http://www.faqs.org/rfcs/rfc2401.html>.

Projekty, programy:

[11] *BIND* Verze 9.4.2 [online].
Dostupné z URL: <http://www.isc.org/>.

[12] *FreeS/WAN* [online].
Dostupné z URL: <http://www.freeswan.org/>.

[13] *Iperf*. Verze 1.7.0 [online].
Dostupné z URL: <http://dast.nlanr.net/Projects/Iperf/>.

[14] *NAT32*. Verze 1.8 [online].
Dostupné z URL: <http://www.nat32.com>.

[15] *OpenSSL*. Verze 0.9.8g [online].
Dostupné z URL: <http://www.openssl.org>.

[16] *OpenVPN*. Verze 2.0.9 [online].
Dostupné z URL: <http://openvpn.net/>.

[17] *SoftPerfect Bandwidth Manager*. Verze 2.7 [online].
Dostupné z URL: <http://www.softperfect.com/products/bandwidth/>.

[18] *WinpkFilter*. Verze 3.0.4 [online].
Dostupné z URL: <http://www.ntkernel.com/w&p.php?id=7>.

[19] *Wireshark*. Verze 1.0.0 [online].
Dostupné z URL: <http://www.wireshark.org/>.

[20] *Virtual PC 2007*. [online].
Dostupné z URL: <>.

[21] *VMware Server*. Verze 1.0.4 [online].
Dostupné z URL: <http://www.vmware.com/products/server/>.

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

AH	Authentication Header
3DES	Triple DES
DES	Data Encryption Standard
DETEC	Decentralized End-To-End Communicator [1]
ESP	Encapsulating Security Payload
HMAC	The Keyed-Hash Message Authentication Code
IP	Internet Protocol
IPsec	IP security
MTU	Maximum Transmission Unit
NAT	Network address translation
PST	Protocol for Secure Tunnels [1]
SSL	Secure Sockets Layer
TLS	Transport Layer Security
VPN	Virtual Private Network

SEZNAM PŘÍLOH

A OBSAH PŘILOŽENÉHO CD-ROM

Struktura adresářů:

- **program** – obsahuje vytvořený kód v rámci implementace systému DETEC, v programovacím jazyce C++ ve vývojovém prostředí (IDE) Microsoft Visual Studio 2008 (Visual C++).
  - **Client** – kód klienta
  - **Server** – kód spojovacího serveru
- **text** – obsahuje text závěrečné zprávy diplomové práce
  - **pdf** – obsahuje PDF verzi textu závěrečné zprávy diplomové práce
  - **zdrojove** – obsahuje zdrojové soubory závěrečné zprávy diplomové práce vytvořené pro sázecí systém LaTeX