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

# ANONYMNÍ POHYB V SÍTI INTERNET

ANONYMOUS COMMUNICATION ON THE INTERNET

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE Bc. JAN HOŘEJŠ
AUTHOR

VEDOUCÍ PRÁCE Ing. MARTIN ROSENBERG
SUPERVISOR

BRNO 2014

**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

**Fakulta elektrotechniky a komunikačních technologií**

**Ústav telekomunikací**

# Diplomová práce

magisterský navazující studijní obor
**Telekomunikační a informační technika**

| | | | |
|---|---|---|---|
| ***Student:*** | Bc. Jan Hořejš | ***ID:*** | 129352 |
| ***Ročník:*** | 2 | ***Akademický rok:*** | 2013/2014 |

**NÁZEV TÉMATU:**

## Anonymní pohyb v síti internet

**POKYNY PRO VYPRACOVÁNÍ:**

Cílem diplomové práce bude popsat současné možnosti anonymního pohybu v síti internet, testování a měření vybraných řešení a určení možných slabin, které by mohli vést k odhalení identity komunikující strany, případně serveru.
Student se zaměří na konfiguraci sítě TOR, konfiguraci skryté služby (Hidden Service) v síti TOR a určí míru zabezpečení a anonymity dané konfigurace. Výstupem diplomové práce bude návrh a konfigurace modelu sítě a analýza možností, jak odhalit identitu uživatelů a serverů (možnosti diskreditace Hidden Service). Student se zaměří i na možnosti nasazení anonymní autentizace v anonymní síti TOR. Výsledkem bude návrh řešení pro kompletně anonymní pohyb a autentizaci v síti internet.

**DOPORUČENÁ LITERATURA:**

[1] ACQUISTI, Alessandro. Digital privacy: theory, technologies, and practices</i>. New York: Auerbach Publications, c2008, xxii, 470 p. ISBN 14-200-5217-9.

[2] BAILEY, Matthew. Complete guide to internet privacy, anonymity. S.l.: A Nerel publication. ISBN 3950309306.

| | | | |
|---|---|---|---|
| ***Termín zadání:*** | 10.2.2014 | ***Termín odevzdání:*** | 28.5.2014 |

***Vedoucí práce:*** Ing. Martin Rosenberg
***Konzultanti diplomové práce:***

**doc. Ing. Jiří Mišurec, CSc.**
*Předseda oborové rady*

**UPOZORNĚNÍ:**

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

## Abstrakt

Cílem diplomové práce bylo popsat současné možnosti anonymního pohybu po Internetu. Teoretická část se zaměřuje na tři hlavní způsoby anonymizace se zaměřením na síť Tor. V práci jsou popsány výhody a nevýhody jednotlivých řešení a možné útoky na ně. V další části je demonstrována síť Tor, implementace Skryté služby, zabezpečeného anonymního přístupu uživatelů k serveru a možných útoků vůči tomuto návrhu. Práce zároveň obsahuje výsledky měření všech tří způsobu anonymizace a vlivů na jejich rychlost.

## Klíčová slova

Tor, Cibulové směrování, Skrytá služba, VPN, Virtuální privátní síť, Proxy server, Apache, Certifikát, Certifikační autorita, Anonymita, Anonymizer, ICMP, Vidalia, Diffie-Hellman, Člověk uprostřed, DoS, DDoS, CGI, OpenSSL, X.509, SSL, HTTPS, HTTP, RSA, TDES

## Abstract

The objective of this master's thesis was to describe current capabilities of anonymous browsing over the Internet. The theoretical part focuses on three main methods of anonymization with main focus on Tor network. The master's thesis describes advantages and disadvantages of different solutions and possible attacks on them. In the next part is demonstrated Tor network, implementation of Hidden service and secured access to the server for clients and possible attacks against this proposal. The work also includes the results of measurements of all three anonymizers and the effects on their speed.

## Key words

Tor, The Onion routing, Hidden service, VPN, Virtual private network, Proxy server, Apache, Certificate, Certification authority, Anonymity, Anonymizer, ICPM, Vidalia, Diffie-Hellman, Man in the middle, DoS, DDoS, CGI, Open SSL, X.509, SSL, HTTPS, HTTP, RSA, TDES

HOŘEJŠ, J. *Anonymní pohyb v síti internet.* Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2014. 82 s. Vedoucí diplomové práce Ing. Martin Rosenberg.

## Prohlášení

Prohlašuji, že svoji diplomovou práci na téma Anonymní pohyb v síti internet 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 tohoto projektu 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 28.5.2014 .....

Bc. Jan Hořejš

## Poděkování

Děkuji vedoucímu diplomové práce Ing. Martinu Rosenbergovi za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce.

V Brně dne 28.5.2014 .....

Bc. Jan Hořejš

Faculty of Electrical Engineering
and Communication
Brno University of Technology
Technicka 12, CZ-61600 Brno
Czech Republic
http://www.six.feec.vutbr.cz

## PODĚKOVÁNÍ

Výzkum popsaný v této diplomové práci byl realizován v laboratořích podpořených z projektu SIX; registrační číslo CZ.1.05/2.1.00/03.0072, operační program Výzkum a vývoj pro inovace.

Brno ..... .....
(podpis autora)

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY

EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI

2007–13
OP Výzkum a vývoj pro inovace

# OBSAH

**SEZNAM OBRÁZKŮ** ..... **6**

**SEZNAM TABULEK** ..... **6**

**SEZNAM GRAFŮ** ..... **6**

**1. ÚVOD** ..... **9**

1.1. Anonymita ..... 10

1.1.1. Výhody anonymity ..... 10

1.1.2. Nevýhody anonymity ..... 11

1.2. Anonymita na internetu ..... 11

**2. Řešení anonymizace v Internetu** ..... **12**

2.1. Proxy servery ..... 12

2.2. Virtuální privátní síť ..... 14

2.2.1. Princip navázání spojení ..... 15

2.3. Síť Tor ..... 17

2.3.1.Tor a jeho historie ..... 17

2.3.2. Princip ..... 19

2.3.3. Návrh Toru ..... 21

2.3.4. Skrytá služba (Hidden service) ..... 25

2.3.5. Útoky na síť Tor ..... 28

2.3.6. Útoky na Skrytou službu ..... 36

2.4. Anonymní autentizace vůči serveru ..... 42

2.4.1. Zabezpečení přenosu prostřednictvím protokolu HTTPS ..... 42

2.4.2. Certifikační autorita ..... 43

2.4.3. TLS/SSL ..... 44

**3. Implementace anonymního přístupu** ..... **50**

3.1. Nastavení ICMP zpráv ..... 50

3.2. Instalace Toru ..... 51

3.2.1. Druhy balíčků ..... 51

3.2.2. Získání balíčku Tor ..... 52

3.2.3. Ověření obdrženého balíčku ..... 53

3.2.4. Spuštění Toru ..... 54
3.3. Instalace skryté služby ..... 56
3.4. Implementace zabezpečeného anonymního přístupu pomocí HTTPS .... 57
3.5. Výsledky měření ..... 71
**4. ZÁVĚR ..... 73**
**LITERATURA ..... 74**
**SEZNAM ZKRATEK ..... 76**
**SEZNAM PŘÍLOH ..... 78**
**OBSAH PŘILOŽENÉHO DVD ..... 82**

# SEZNAM OBRÁZKŮ

Obr. 2.1. Princip směrování skrze proxy server ..... 12
Obr. 2.2. Princip komunikace skrze VPN tunel ..... 14
Obr. 2.3. Přenos dat sítí Internet ..... 19
Obr. 2.4. Dotaz uživatele na databázi Toru ..... 20
Obr. 2.5. Komunikace uživatele skrze síť Tor ..... 20
Obr. 2.6. Struktura řídící a okruhové buňky ..... 22
Obr. 2.7. Princip vytvoření okruhů a komunikace s cílovým serverem. ..... 23
Obr. 2.8. Rušení části okruhového spojení ..... 24
Obr. 2.9. Počáteční sestavení okruhů pro skrytou službu ..... 26
Obr. 2.10. Navázání spojení skrze Bobovy body pro seznámení (IPx) ..... 27
Obr. 2.11. Navázané spojení mezi uživatelem a skrytou službou ..... 28
Obr. 2.12. Útok typu Člověk uprostřed ..... 32
Obr. 2.13. Útok odstřelovače ..... 39
Obr. 2.14. Útok na Skrytou službu ..... 41
Obr. 2.15. Navázaní TLS/SSL spojení ..... 45
Obr. 2.16. Jednosměrná SSL autentifikace ..... 47
Obr. 2.17. Obousměrná SSL autentifikace ..... 48
Obr. 2.18. Útok hearbleed ..... 49
Obr. 3.1. Nastavení firewallu systému Windows ..... 50
Obr. 3.2. Tělo zprávy pro získání Toru skrze email ..... 52
Obr. 3.3. Instalace programu Gpg4win ..... 53
Obr. 3.4. Naimportování klíče pro ověření podpisu balíčku Tor ..... 54
Obr. 3.5. Porovnání klíče a digitálního podpisu aplikace ..... 54
Obr. 3.6. Hlavní okno programu Vidalia ..... 55
Obr. 3.7. Nastavení programu Vidalia ..... 55
Obr. 3.8. Nastavení informací o serveru ..... 56
Obr. 3.9. Importování kontejneru do prohlížeče Toru ..... 65
Obr. 3.10. Nastavení důvěry pro kořenovou CA ..... 66
Obr. 3.11. Informace o serveru a platformě OS ..... 68
Obr. 3.12. Výpis CGI skriptu ..... 70

# SEZNAM TABULEK

Tab. 2.1. Zkažených cibulových směrovačů a druhy útoků ..... 34
Tab. 3.1. Výsledné hodnoty průměrných rychlostí v závislosti na druhu spojení a velikosti souborů ..... 71

# SEZNAM GRAFŮ

Graf 2.1. Závislost počtu výstupních směrovačů v čase ..... 32
Graf 3.1. Závislosti průměrné rychlosti na lokalitě a způsobu připojení ..... 72

# 1. ÚVOD

V dnešní moderní době, která je tvořena díky pokroku stále novými a novými technologiemi, dochází k pomalé změně našich základních návyků. Jedním z těchto návyků je vzájemná komunikace a to jak mezi lidmi, tak i mezi lidmi a zařízeními.

Rozmach Internetu, který se dá hodnotit jako jeden z největších objevů minulého století, podnítil vznik nových způsobů, jakým komunikovat s vnějším světem. V dřívějších dobách se zprávy předávaly ústně, pomocí dopisů, telegrafů a mnohých jiných. Tyto způsoby měly mnohé nevýhody, avšak jednu velkou výhodu. Tou je poměrně dostatečná kontrola nad tím, jaké informace sdělujeme druhé straně a kdo se k nim může v průběhu komunikace dostat.

Dnes, právě díky internetu, má většina lidí milný pocit, jestliže není člověk viděn, tak o něm nikdo nic neví. Dále také dochází ke ztrátě zábran, které při osobním kontaktu udržuje. Z pocitu kontroly a přehledu, vědomě a hlavně i nevědomě prozrazuje okolí velké množství informací, které by za normálních okolností nebyly vyzrazeny.

Hlavní přínos této práce spočívá v umožnění komunikace uživatelům, v rámci internetu tak, aby je nebylo možné identifikovat. Tím bude docíleno anonymní komunikace, která může být využita například k šíření informací, které by byly normální cestou cenzurovány a které by mohly vést k postihu uživatele, který se snaží využít jednoho ze svých základních práv. Dalším přínosem této práce je zmapování různých způsobů, jak docílit anonymity a vybrání vhodného zabezpečení komunikace.

Zbytek této práce je členěn následovně. V první kapitole jsou popsány důvody a smysl anonymní komunikace v rámci internetu. Ve druhé kapitole jsou rozebrány způsoby, jakým lze anonymizovat pohyb po internetu. Následně se práce zabývá nejvhodnějším protokolem Tor, jeho principem a možnými útoky vůči němu a Skryté službě. Poslední část kapitoly se zaobírá možnou ochranou a anonymním autentizací klienta vůči serveru. Čtvrtá kapitola pojednává o implementaci samotné sítě Tor, vytvořením a nakonfigurováním Skryté služby, za účelem zabezpečeného přístupu k serveru. Následně jsou implementována dodatečná zabezpečení, které zároveň poskytují i anonymní autentizaci přistupujícího uživatele.

## 1.1. Anonymita

Původ slova anonymita pochází z řecké slova anōnymos znamenající „bezejmenný“ nebo „nepodepsaný“. Význam tohoto slova se vztahuje k osobě, u které neznáme její identitu a ze které nejsme schopni určit, o koho se jedná [14].

Utajení totožnosti neboli anonymita není žádným novým objevem, který se začal šířit s nástupem Internetu. Anonymita je používána po staletí k šíření důležitých informací, aniž by byl autor identifikován. Důvody k tomuto chování mohou být různé, především se jedná o odsuzování a předsudky společnosti vůči dané osobě. Může se jednat o politické, genderové, náboženské a mnohé jiné předsudky, které mohou předávanou informaci zdiskreditovat, a tím snížit její vypovídající hodnotu. Použití anonymity tak ochraňuje tuto osobu před možnou persekucí i možnými sankcemi.

Z těchto důvodů se s anonymitou spojuje další slovo, a tím je pseudonym. Pseudonym vytváří falešnou identitu uživatele, která umožňuje komunikovat s možností být identifikován, avšak bez možnosti zjištění pravé identity uživatele. Nejznámějším použitím pseudonymů jsou příkladem jména knižních autorů, herců a mnohá jiná.

V posledních letech, s narůstajícím počtem nových uživatelů internetu, kteří využívají práva na svobodný projev, dochází k dohadům, zdali je anonymita dobrá či špatná. Je nutné si uvědomit, že nic není černobíle a vše má svá pro a proti.

### 1.1.1. Výhody anonymity

- Umožnění práva na svobodný projev osobám, které jsou umlčovány za pomoci cenzury. Politické režimy snažící se vyvíjet aktivitu proti těmto osobám za účelem umlčení a potrestání antipolitických názorů.
- Výhod anonymity může být použito k nahlášení nelegálních či potencionálně trestných aktivit. Takovéto anonymní tipy mohou být použity k informování veřejnosti o těchto praktikách, za pomoci novinářů či policejních složek.
- Možnost otevřeně diskutovat nad citlivými tématy, které by za normálních okolností byly zahanbující. Je možné otevřeně diskutovat nad sexuálními problémy, zdravotními, náboženskými a mnohými jinými.
- Schopnost nezaujatě posuzovat a to s větší upřímností, jelikož nejsou předem vytvářeny předsudky, které by ovlivnily naše rozhodování.
- Seberealizace sebe sama s následným rozvojem svojí osobnosti.

### 1.1.2. Nevýhody anonymity

Mnohé otázky, které napadnou člověka při slově anonymita jsou: „Proč nechce prozradit svoji totožnost? Skrývá něco? Dělá něco nelegálního?“

Pomocí anonymity může docházet k:

- Páchání trestné činnosti. Pomocí anonymní komunikace mohou být řízeny různé obchody se zbraněmi, lidmi, drogami a to společně s internetovou kriminalitou v podobě krádeží identit, kreditních karet a mnoha dalších.
- Poškozování majetku a protivládním aktivitám, například k plánování teroristických útoků a šíření teroru proti jednotlivcům nebo skupinám.
- Podpoře sexuálních úchylek, které mohou být nebezpečné pro společnost, například šíření pedofilie.
- Pronásledování osob skrze internet a následného šikanování tzv. kyberšikana.
- Nemožnost ověření relevantnosti zmíněných tvrzení.

Vše má svá negativa, která často převyšují i majoritní zastoupení pozitivních vlastností. Na toto téma se v posledních letech, jak už bylo zmíněno, vedou sáhodlouhé diskuze. Z následujících kapitol bude patrné, že samotné vlády, které podporují demokracii a šíření informací, zároveň brojí proti anonymitě, která zároveň představuje hrozbu.

## 1.2. Anonymita na internetu

I přes snahu být na internetu anonymní, a to i za pomocí pseudonymů, nejsme tohoto cíle schopni dosáhnout vždy na 100%. Internet je koncipován tak, že vždy musí být identifikován zdroj a cíl komunikace tak, aby bylo možné předávat data mezi těmito dvěma body.

Příkladem této komunikace může být uživatel, který pod smyšleným pseudonymem přispěje do diskuzního fóra. Z první pohledu se zdá, že tento příspěvek byl učiněn anonymně, avšak správce serveru zná identifikační údaje o uživateli, v podobě uživatelského jména, často emailové adresy a především IP adresy, ze které byl příspěvek přidán. Pomocí této adresy je, za určitých okolností, možné dopátrat konkrétní počítač a následně i osobu, která jej v danou dobu použila.

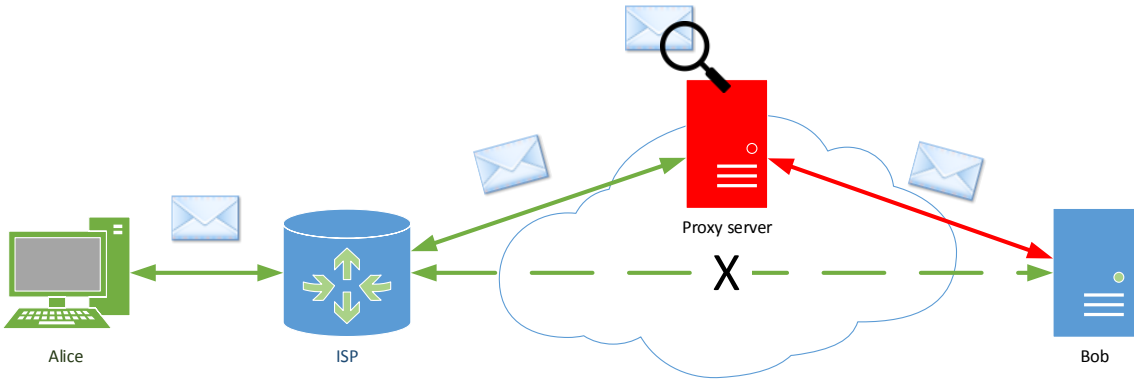
Konkrétní postupy, jak zjistit a také zamezit zjištění identity budou probrány v následujících kapitolách, jelikož lze k anonymitě na internetu přistupovat různými způsoby.

# 2. Řešení anonymizace v Internetu

V následujícím textu budou popsány tři nejrozšířenější techniky, jakými se uživatelé internetu snaží dosáhnout anonymního a tedy nevystopovatelného pohybu v internetu. Budou zde probrány vlastnosti proxy serverů, privátních virtuálních sítí a podrobně popsán projekt Tor, sloužící k efektivní anonymizaci, zároveň s cílem uživatelské jednoduchosti.

## 2.1. Proxy servery

Jedním z nejjednodušších a nejrozšířenějších způsobů jakým uživatelé anonymizují svůj pohyb po Internetu je za pomoci proxy serveru. Ten využívá jednoduchého modelu komunikace, kdy uživatel využívá prostředníka ke komunikaci s koncovým zařízením. Tento prostředník může být fyzické zařízení nebo program. Klient vytvoří dotaz, například na webovou stránku, přenos souboru či jiné a tento dotaz zašle proxy serveru. Ten zprávu modifikuje a tu následně zašle na cílovou stanici. Cílový server zpracuje požadavek a následně jej navrátí proxy serveru, který zprávu přepošle zpět klientovi.




**Obr. 2.1. Princip směrování skrze proxy server**

Proxy servery patří k jedněm z nejrozšířenějších služeb v rámci Internetu. Jejich možné nasazení je velmi rozmanité. Proxy servery mohou být využívány ke směrování specifických požadavků na příslušné servery, a tím vytvoří přístupové brány k různým službám. Ve školách a pracovištích mohou být proxy servery využity k omezení a filtrování služeb. Dále také k vytvoření vyrovnávací paměti, kdy je možné předčítat obsah webových stránek nebo DNS záznamů, avšak nejvíce známým využitím proxy serverů jsou tzv. anonymizéry neboli anonymní proxy servery či webově orientované proxy servery.

Cílem anonymních proxy serverů je poskytnout uživateli přístup k cílovému serveru. Ten může být nějakým způsobem blokován, buď z důvodů geografických či

politických. Zároveň se snaží poskytnout jistou míru anonymity tím, že skryje informace o tomto uživateli.

Rozdělit anonymní proxy servery lze do tří základních skupin, které se liší v zacházení s přenášeným daty.

- Anonymní proxy – Je nejčastěji používaným typem. Sám sebe identifikuje jako proxy server a poskytuje jistou míru anonymity pro uživatele. Je často přístupný bez poplatků, avšak kvůli velkému využití ostatními uživateli je také dosti pomalý.
- Matoucí proxy – je podobný předcházejícímu typu serveru, avšak pozměňuje hlavičku HTTP a mění v ní zdrojovou IP adresu určenou pro identifikace zdroje komunikace.
- Vysoce anonymní proxy – Téměř ve všech případech je tato služba placená. Proxy v tomto případě není veřejně známá a tím pádem ji není možné identifikovat ze strany serveru. Takto přistupující klient, za pomoci vysoce anonymního proxy, bude považován za normálního uživatele.

Implementace na straně klienta je velmi jednoduchá. Seznam serverů je dostupný například zde [18]. Všechny internetové prohlížeče jsou vybaveny možností připojení k Internetu za pomoci proxy serveru. Stačí pouze zadat příslušnou IP adresu serveru a port.

I přes to, že jsou anonymizéry jednou z nejrozšířenějších služeb, jsou zároveň velmi náchylné ke zneužití. Existuje mnoho bezpečnostních rizik, která mohou odhalit uživatele a jeho komunikaci.

Největší riziko spočívá v samotném principu fungování proxy serverů. Všechna data, která jsou odeslána na cílovou stanici, musí projít skrze proxy server a to často v nezašifrované formě. Jelikož nám není známo, kdo vlastní a provozuje tento server, často známe pouze IP adresu a port, je možné, že tento server zachytává veškerá data a analyzuje je. Útočnici se často snaží nalákat uživatele k velmi rychlé proxy, která funguje zdarma s cílem získat data, která následně zneužijí.

Dalším závažným problémem je možná vystopovatelnost výchozí stanice. Tím, že proxy server komunikuje s cílovým server, získává cílová stanice adresu tohoto serveru. Adresa může být použita k získání informací o majiteli serveru, který může být donucen k vydání totožnosti zdrojové IP adresy použité pro komunikaci. Samozřejmě je možné použít několikanásobné přesměrování, avšak informace o průchozích proxy nám nejsou známy a mohou představovat další rizikové body v komunikačním řetězci.

Volně dostupné anonymní proxy se také mohou zachovat jako transparentní proxy a předat zdrojovou adresu uživatele cílovému zařízení i přes to, že slibovalo, že se tak nestane.

V neposlední řadě je zde problém s nedostupností. Volně dostupné anonymní proxy servery jsou veřejně známé. Toho mohou využít koncové stanice tím, že porovnají seznam veřejně dostupných proxy se stanicí, která se snaží na ně přistoupit a v případě shody tuto komunikaci odmítnout. Stejného způsobu může využít i ISP a zablokovat komunikaci na tento server.

## 2.2. Virtuální privátní síť

Virtuální privátní síť (VPN – Virtual Private Network) je jednou z další možností anonymizace uživatele v Internetu. VPN využívá principu okruhového spojení, kdy vytváří virtuální spojení vícero fyzicky vzdálených počítačů tak, že se chovají, jakoby byly součástí přímo připojené lokální sítě.

**Obr. 2.2. Princip komunikace skrze VPN tunel**

VPN se dělí na dva typy dle použitého přístupu:

- Místo-Místo VPN – spojuje dvě odlehlé sítě dohromady skrze Internet. Využívá se například k propojení několika poboček mezi sebou. Jako hraniční body se často používají speciální směrovače, firewally nebo servery, které se starají o spojení. Uživatelé uvnitř sítě nemusí mít žádný speciální program pro komunikaci se vzdálenými stanicemi.
- Vzdálený přístup skrze VPN – tento způsob propojení slouží ke spojení uživatele ke vzdálené privátní síti. Klient musí mít nainstalovaný speciální program, který mu umožní vzdálenou komunikaci skrze Internet.

Další možnosti dělení sítí VPN jsou podle použitého bezpečnostního protokolu:

### IPsec VPN:

Neboli Internet Protocol Security je rozšiřující protokol síťové vrstvy. Je součástí IPv6 protokolu a je používán, jako rozšíření protokolu IPv4. IPsec slouží k propojení dvou koncových zařízení pomocí vzájemné autentifikace a výměny kryptografických klíčů. IPsec zabalí a zašifruje celý paket do nového a opatří jej novou IP hlavičkou sloužící pro komunikaci v nezabezpečené síti.

Nevýhodou tohoto protokolu je, že při průchodu skrze NAT je modifikována hlavička. To má za následek nesouhlas haše celého paketu. Z tohoto důvodu je nutné používat NAT-T, který celý IPsec paket zabalí do další vrstvy, v podobě UDP datagramu, jehož je možné po cestě modifikovat.

### SSL VPN:

Využívá protokolu SSL (Secure Sockets Layer), který se nachází v Relační vrstvě OSI modelu. Pro zašifrování dat je použita autentifikace a asymetrické šifrování, který využívá principu veřejného a privátního klíče. Data jsou směrována skrze HTTPS na port 443. Díky tomu je zaručen bezproblémový průchod skrze NAT a Firewall. Další předností tohoto protokolu je, že není nutný žádný speciální software, jelikož SSL/HTTPS je podporován všemi webovými prohlížeči.

Nevýhodou SSL VPN je, že je určen pro jednotlivé aplikace, a tudíž nemá podporu přístupu do celé sítě jako IPsec. Pro komunikaci, skrze tento protokol, je tedy nutné nakonfigurovat všechna spojení, která budou využívána.

## 2.2.1.Princip navázání spojení

Navázání zabezpečeného spojení skrze VPN je složitější, než skrze jednoduchý webový anonymizér.

1. Klient se připojí k veřejně dostupné VPN bráně a identifikuje se.
2. Server ověří totožnost. V případě úspěchu, sestaví virtuální okruh a přiřadí klientovi IP adresu vnitřní sítě.
3. Klient následně může požadovat komunikaci s další stanicí v privátní stanici. Vytvoří požadavek na cílovou stanicí, uvede svoji privátní IP adresu, celý paket zabalí, pomocí protokolu IPsec, a ten následně odešle s veřejnou IP adresou na VPN bránu.
4. Brána po přijetí zprávu dešifruje a zašle paket s privátními adresami v hlavičkách na cílovou stanici.

5. Cílová stanice po obdržení zprávy odpoví na cílovou privátní adresu VPN klienta a odešle jej.
6. VPN brána přijatý paket zabalí a přiřadí cílovou veřejnou IP adresu a odešle.
7. Klient přijme paket a dešifruje jej.

Z výše zmíněného principu fungování VPN je možné dedukovat použití pro anonymizaci v Internetu. Veškerá komunikace od uživatele až po VPN bránu je šifrovaná, a tím pádem chráněna před ISP nebo útočníkem. Zároveň vytváří virtuální tunelové spojení, kterým lze obejít restrikce v připojení k Internetu. Dále nám umožňuje získat novou IP adresu, kterou lze použít pro komunikaci. To má za následek skrytí identity a místa původu.

I přes to, že je tento způsob spojení mnohonásobně bezpečnější, než použití anonymizéru, má i VPN několik slabin.

Předně je nutné si uvědomit, že VPN spojení, oproti proxy serverům, není poskytováno zdarma. Pokud tedy chceme využívat VPN, je nutné jej zakoupit. To samo o sobě představuje dostatečné riziko pro naši anonymitu. Při koupi je nutné zmínit své jméno, emailovou adresu a zaplatit, například pomocí kreditní karty. Tento problém se dá částečně vyřešit používáním omezených verzí programů pro VPN, které nemají kompletní podporu pro zabezpečení, mají nízkou přenosovou rychlost, ale není nutné uvádět citlivá data. Další možností je využití otevřené peněžní měny Bitcoin pro anonymní platby.

Připojení pomocí VPN taktéž snižuje přenosovou rychlost, jelikož je nutné připojit se k jednomu přístupovému bodu v Internetu a zároveň je nutné přenést velké množství režijních dat.

Dalším problémem je, že útočník sledující náš datový tok je schopen determinovat, že dochází ke komunikaci mezi námi a přístupovým bodem VPN. Útočník není sice schopen zjistit obsah zpráv, ale již znalost tohoto datového toku může být dostačující k tomu, aby bylo zaútočeno na tento bod, například pomocí DoS útoku, případně může být provozovatele VPN donucen vydat informace o uživateli.

## 2.3. Síť Tor

### 2.3.1. Tor a jeho historie

Síť Tor [24] (dříve TOR je zkratkou z The Onion Router – Cibulové směrování) je volně šiřitelný software, který se za pomoci síťových tunelů snaží zabezpečit a anonymizovat pohyb uživatele na Internetu tak, aby nedošlo k narušení jeho soukromí. K tomu využívá síť přístupových bodů, tvořených dobrovolníky po celém světě, k vytvoření vstupně/výstupních bodů mezi uživateli nebo servery, a tím k zamezení pozorování a analýze této sítě.

Cílem tohoto projektu je potlačení sledování a následné persekuce osob, které potřebují skrýt svoji identitu a svoji lokaci. Síť je především určena pro disidenty, žurnalisty, oznamovatele a jiné osoby, které nemají přístup ke svobodným informacím z důvodu blokace třetích stran.

Tor je díky své anonymitě velmi kontroverzním projektem, který na jedné straně chrání lidská práva, za pomoci anonymizace, na druhou stranu je tato síť využívána i pro nelegální činnosti, jako černý obchod s osobními informacemi, drogami, podporou terorizmu a mnoho dalších kriminálních aktivit, které jsou společensky nepřípustné. V dnešní moderní internetové době je nasnadě i šíření autorsky chráněných děl za využití Torrentů, které pracují na principu komunikace klient-klient (P2P). Avšak i když je možné takto využít síť Tor, dochází k tomuto využití minoritně. Tor je tvořen velkým počtem dobrovolníků, kteří poskytují své internetové připojení, avšak uživatelů Toru je mnohonásobně více, a tím dochází k velmi nízkým přenosovým rychlostem, které se dají srovnat s vytáčeným připojením. Jak již bylo zmíněno, síť je tvořena dobrovolníky, kteří při poskytnutí svého internetového spojení riskují postih v případě, že někdo využije jejich počítač k některým nelegálním činnostem. I přes všechna negativa této sítě, která vytváří dojem, že Tor je jedna velká nelegální síť, převažují její pozitiva. Kontroverzi projektu také potvrzuje následující odstavec, z důvodu financování americkou vládou. Ta zároveň sama útočí na tuto síť a snaží se ji prolomit. [1]

Historie Cibulové sítě se datuje do roku 1995 [20], kdy započaly práce pod vedením americké ONR (Office of Naval Research), která spadá pod Námořnictvo Spojených států amerických (US Navy) s cílem zabezpečit vládní komunikaci. První demonstrace sítě byla prezentována 31. dubna 1996 na Prvním semináři o skrývání informací, kde byla představena síť o pěti uzlech. S rostoucím potencionálem, do projektu v roce 1997 vstupuje Agentura pro výzkum pokročilých obranných projektů (DARPA). V průběhu let dochází k vývoji od verze nula až po verzi jedna. Ta je v roce 2002 opuštěna z důvodu zastaralosti a přebujelosti. Jsou zahájeny práce na druhé verzi, které byly usnadněny prudkým rozvojem Internetu a podporou směrování paketů mezi serverem a klientem pomocí proxy serverů. Díky tomu byla koncem roku 2002 vypuštěna první Alfa verze programu.

[1] Útoky na prolomení sítě Toru nejsou jen ze strany americké vlády. Všechny mocnosti světa, včetně mnoha společností a jednotlivců se snaží prolomit tuto síť a dostat se k původcům datové komunikace.

Následně pokračuje vývoj s podporou EFF (Electronic Frontier Foundation) a k implementaci Skrytých služeb (Hidden services), která přinesla anonymitu i pro weby a servery. Cibulová služba je od té doby rozvíjena jako neziskový vzdělávací projekt Tor, který je z cca 80% financován americkou vládou. Zbytek spadá pod podporu švédské vlády, dalších organizací a dobrovolných dárců.

Hlavním cílem Toru je poskytnout návrh, který bude poskytovat nízkou odezvu na požadavky a zamezí útočníkovi identifikovat uživatele této sítě. S ohledem na tyto cíle, bylo nutné specifikovat několik důležitých bodů, aby bylo možné vytvořit komplexní a dostatečně zabezpečenou síť.

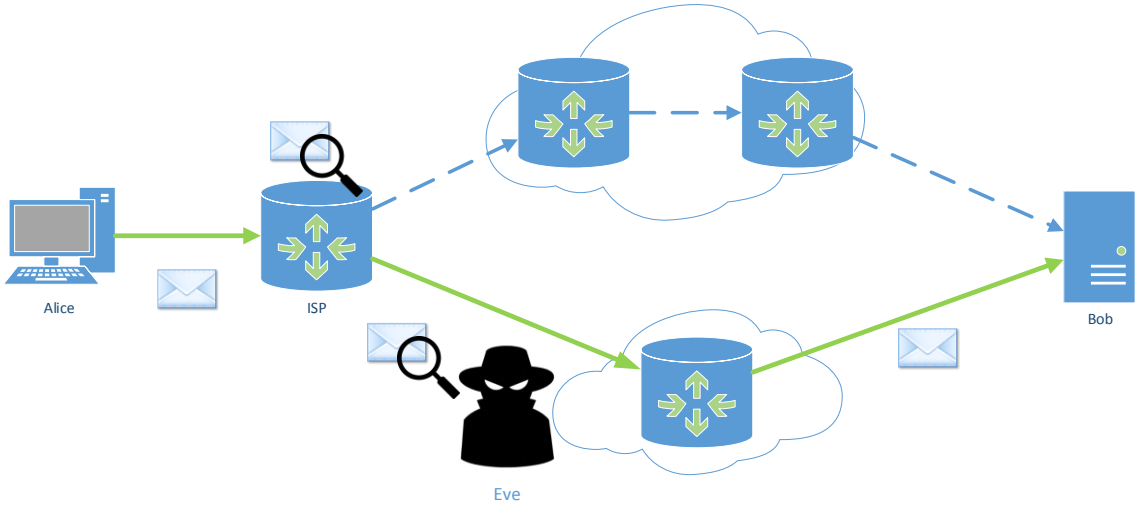
- **Nasazení:** Návrh Toru je určen pro reálné nasazení a provoz. Je tedy nutné, aby nasazení a provoz byly nízkonákladové, síť nevyžadovala větší propustnost, než je nezbytně nutné (redundance, haše atd.) a než jsou dobrovolníci ochotni poskytnout. Složitost implementace musí být jednoduchá, nenáročná a nesmí vyžadovat speciální požadavky u třetích stran, v podobě webových serverů a jiných. Důležité je také zamezit tomu, aby útočníci zahrnuli poskytovatele do nelegálních činností.
- **Použitelnost:** Tor musí být pro uživatelsky orientovaný. To znamená, že musí být velmi jednoduchý pro použití a nebude vyžadovat velkou míru interakce od uživatele a jeho znalostí. V případě nesplnění této podmínky by došlo k nízkému rozšíření služby, což by mělo za následek malý počet uživatelů, které by bylo těžké anonymně skrýt v síti, kde je pouze hrstka uživatelů.
- **Pružnost a jednoduchost:** Protokol musí být jednoduchý, pružný, dobře definovaný a pochopitelný, aby mohl být použit jako referenční pro další vývoj, implementaci a bezpečnostní analýzy.

Tor se dále snaží vyhnout některým návrhům, které ještě nejsou vyřešené nebo jsou řešeny jinde. Dizajn protokolu nepokrývá kompletní zabezpečení a anonymizaci koncových bodů, chrání pouze uživatele v této síti. Dále neposkytuje protokolovou normalizaci, příkladem komunikace přes HTTP protokol, který nechrání před prozrazením informací adresátovi.

Dalším příkladem budiž požadavek na internetovou stránku, kdy je nutné překládat adresu. Tím dochází k prozrazení cíle komunikace DNS serveru. Z těchto důvodů je nutné spojit Tor s dalšími vrstvami, které pomohou zabránit úniku těchto informací. Mezi tyto služby patří například Privoxy a Anonymizery.

### 2.3.2.Princip

Klasické směrování v Internetu, má za cíl najít nejlepší cestu k doručení zpráv z výchozího budu, říkejme tomuto bodu Alice, do cílového bodu pojmenovaného Bob. K tomu slouží nejrůznější směrovací protokoly, které podle kvality linky, propustnosti, počtu skoků a mnoha jiných parametru snaží determinovat nejlepší cestu pro jednotlivé pakety. Pomocí znalosti těchto směrovacích protokolů a znalosti výchozího či cílového bodu, jsme schopni provádět různé typy útoků a analýz. Jednou z těchto analýz je síťová analýza, která monitoruje datový provoz.




**Obr. 2.3. Přenos dat sítí Internet**

Z přenášených dat je útočník či poskytovatel internetu (ISP) viz Obr.2.3. schopen zachytit velké množství informací a to i když jsou data šifrována. Vše je lehce identifikovatelné pomocí obsahu datových paketů, které se skládají z řídících dat, obsahující velké množství užitečných informací, a dále samotných přenášených dat uživatele, která nemusí být primárně cílem.

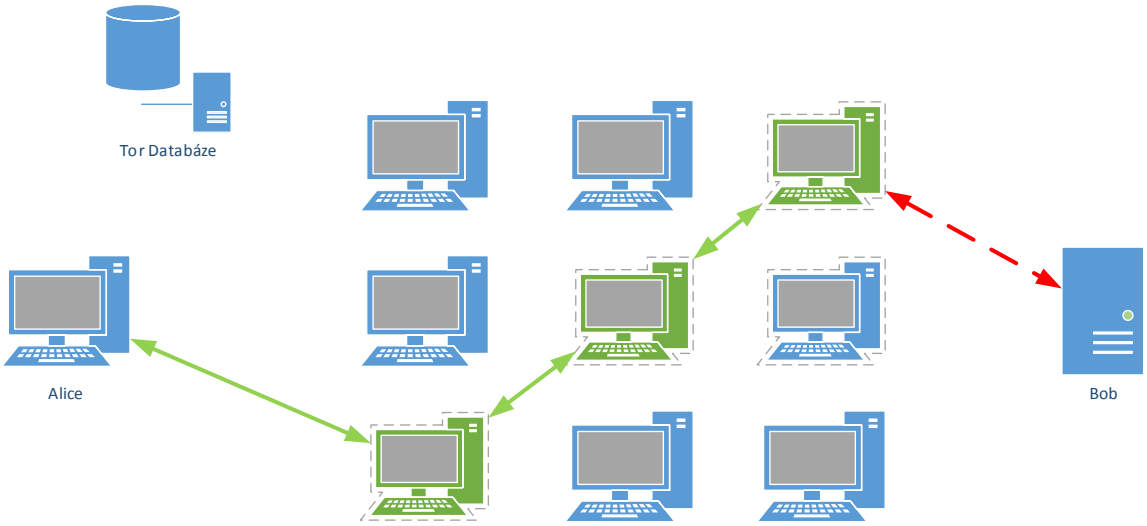
Ze zaznamenaných dat lze následně jednoduše vyčíst, že Alice komunikuje s Bobem a naopak. Samozřejmě toto je pouze jeden ze základních nástrojů pro analýzu, existují také pokročilejší systémy pro monitorování, využívající páteřní sítě, porovnávání aktivit různých uživatelů a pomocí podobného vzorce chování spojovat jednotlivé účastníky komunikace.

Jako řešení pro zamezení sledování a anonymizaci uživatele slouží právě síť Tor, která nevyužívá nejlepší cesty k cíli, ale přeposílá data nahodile skrze síť cibulových směrovačů (Onion Router - OR), které jsou tvořeny dobrovolnými uživateli nebo dedikovanými směrovači, tak aby nebylo jednoduché sledovat datový provoz k cíli.

Komunikace je navázána pomocí hostitelského programu zvaném Onion Proxy (OP), který nejdříve pomocí zabezpečené linky kontaktuje databázi Toru, která mu prozradí lokalitu přístupových bodů v podobě OR.

**Obr. 2.4. Dotaz uživatele na databázi Toru**

OP po obdržení informací zahájí komunikaci a vytvoří nahodilá okruhová spojení mezi jednotlivými OR. Poslední bod komunikace ze sítě Tor, slouží jako výstupní brána pro datovou komunikaci, ze které jsou následně zasílána data do cílové stanice (Bob). Bob nemá tušení, že komunikace probíhá právě s Alicí, ale předpokládá, že dotaz byl zaslán od posledního OR. Průběh této komunikace je reprezentován na Obr. 2.5.




**Obr. 2.5. Komunikace uživatele skrze síť Tor**

### 2.3.3.Návrh Toru

Jak již bylo v předchozích odstavcích zmíněno, síť Tor je tvořena uživatelskou aplikací OP, která se stará o získání informací o OR. Vytváří okruhy mezi jednotlivými OR a umožňuje přístup klientským aplikacím do sítě.

Druhým článkem sítě je cibulový směrovač, který zpracovává požadavky od OP, spojuje jednotlivé body komunikace mezi sebou a udržuje zabezpečené spojení mezi jednotlivými směrovači pomocí protokolu TLS. Každý OR uchovává dlouhodobě identifikační klíč, který slouží k podepsání TLS certifikátu. Certifikát je následně použit k podepsání informací o směrovači, který obsahuje adresu, propustnost, klíče a jiné informace potřebné pro databázi Toru. Další klíč, který je použit na OR, je krátkodobý cibulový klíč, který se vytváří při navazování spojení mezi OP a jiným směrovači k přenosu uživatelských dat.

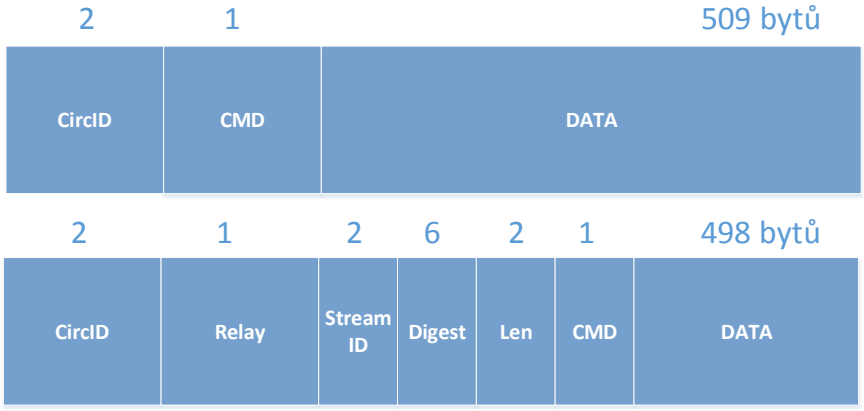
Jak je patrné z výše popsaného textu, ke komunikaci mezi všemi prvky je vždy použita zabezpečená komunikace právě pomocí TLS s krátkodobou životností. To zamezuje útočníkovi modifikovat nebo podvrhnout vyměňovaná data v přijatelném časovém horizontu [5].

#### Buňky:

Při komunikaci v síti jsou použity buňky o pevné velikosti 512 bytů, které obsahují identifikaci okruhu (circID). Tato hodnota specifikuje, ke kterému okruhu buňka náleží, jelikož je možné přenášet více okruhů skrze jedno TLS spojení. Pro každé spojení mezi OP-OR nebo OR-OR je vytvořena unikátní hodnota circID.

Součástí hlavičky je příkaz (CMD), který specifikuje, co se má udělat s přenášenými daty. Toto pole také dělí buňky na dva druhy a to na řídící buňky, které jsou určeny pouze pro další uzel v pořadí a okruhové buňky, které přenášejí uživatelská data. Mezi příkazy této hlavičky patří vycpávka (padding), která se používá k přenosu zpráv udržující TCP spojení (keepalive), vytvořeno (created), vytvoř (create) okruh a v neposlední řadě příkaz znič (destroy), pomocí něhož lze inkrementálně rušit celé spojení se všemi uzly, a tím smazat veškeré stopy po komunikaci.

Součástí buňky okruhu je okruhová hlavička, která je umístěna před samotnými přenášenými daty. Ta obsahuje proudovou identifikaci (StreamID), pro rozlišení vícero proudů. Dále délku záhlaví okruhu, okruhový příkaz a kontrolní součet pro kontrolu celistvosti. Celý obsah hlavičky a přenášených dat je následně zašifrován symetrickou 128 bitovou AES šifrou v režimu čítače (Mění blokovou šifru na proudovou. Další blok je generován správným dešifrováním hodnoty čítače). Mezi okruhové příkazy patří okruhová data (relay data), otevření okruhu (relay begin), zavření okruhu (relay end) při správném ukončení, okruh připojen (relay connected) informuje OP o sestavení okruhu, zrušení okruhu (relay teardown) při poškození okruhu. Dalšími příkazy jsou pro rozšíření (relay extend) a potvrzení (relay extended), že došlo k rozšíření cesty o jeden skok, zrušení a potvrzení části okruhu (relay trunk/trunked) a (relay sendme) pro řízení přetížení.




**Obr. 2.6. Struktura řídící a okruhové buňky**

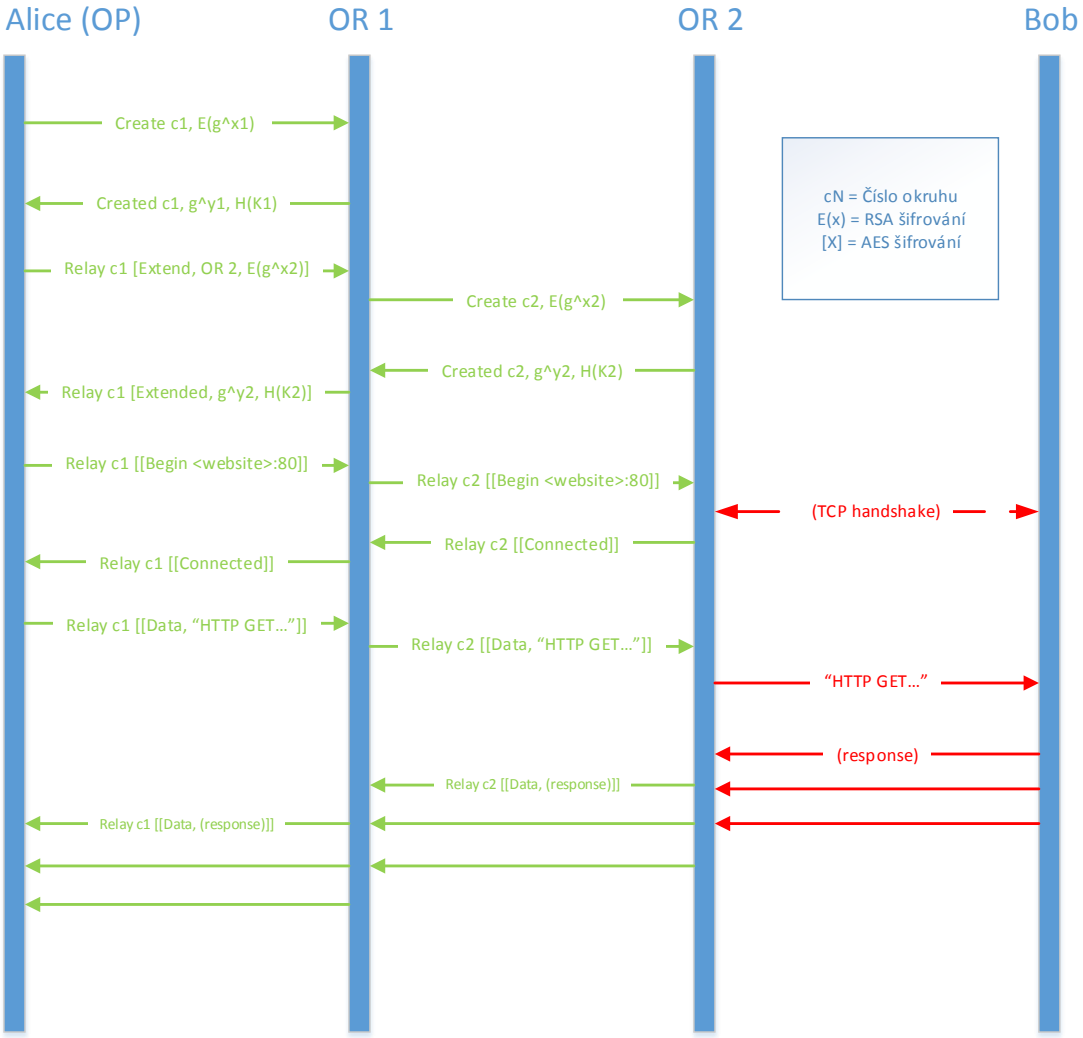
## Okruhy a posílání dat:

Jak již bylo řečeno, okruh může být sdílen vícero TCP spojeními. Jelikož je komunikace mezi jednotlivými body šifrována, bylo by nutné před každým započetím komunikace vytvořit nový a zabezpečený okruh, což by mělo za následek citelné zpoždění a vliv na rychlost. Z tohoto důvodu OP vytváří na pozadí vždy několik okruhů, které jsou připravené k použití. OP také v periodické době tyto okruhy střídá a ty, které byly již využity nebo nebyly použity, zruší a použije čerstvé. Díky tomu nedochází z pohledu uživatele ke zpoždění a poklesu rychlosti.

Navázání spojení je vždy započato ze strany Alice a jejího OP. Ta pošle buňku s příkazem „vytvoř“ prvnímu náhodně zvolenému uzlu (OR 1) a zvolí náhodné číslo okruhu $C_1$. Součástí této buňky jsou také okruhová data, která jsou zašifrována pomocí krátkodobého cibulového klíče RSA, který sdělil OR Tor databázi. V takto zašifrované části hlavičky je obsaženo číslo ($g^{x1}$) pro výměnu symetrického klíče za pomoci Diffie-Hellmanovy výměny klíčů.

Diffie-Hellmanova výměna spočívá v dohodnutí prvočísla z konečné grupy. Toto není nutné, jelikož číslo $g$ již bylo vytvořeno OR a zveřejněno v databázi Toru. OP tudíž může vypočítat své veřejné číslo složené z dohodnutého prvočísla a náhodně vygenerovaného reálného čísla x. Následně dostáváme číslo $A=g^x$ , které je zasláno směrovači zabalené do RSA šifry. Směrovač příchozí šifru RSA dešifruje, zvolí náhodné reálné číslo $y$ a vypočítá z již známého a jím vytvořeného prvočísla $g$ své veřejné číslo $B=g^y$. Z těchto hodnot vytvoří směrovač novou buňku s příkazem vytvořeno a zašle jej zpět Alici se svým veřejným klíčem $B$ a s hašem vyjednaného privátního klíče $K=g^{xy}$.

Po této výměně klíčů jsou obě strany schopny dešifrovat privátní klíč, jelikož klíč Alice $s=B^x$ a zároveň soukromý klíč směrovače $s'=A^y$ jsou totožné. Obě strany mohou zahájit výměnu okruhových buněk.




**Obr. 2.7. Princip vytvoření okruhů a komunikace s cílovým serverem.**

Aby bylo možné dále rozšířit okruhy, vyšle Alice okruhovou zprávu OR 1 s požadavkem „rozšiř“, informacemi, který z uzlů má být kontaktován a zašifrovanými daty pro tento uzel viz Obr.2.7.. Cibulový směrovač po příchodu zprávu dešifruje a vytvoří novou buňku s náhodným číslem okruhu $C_2$, požadavkem „vytvoř“ a zašifrovanými daty od Alice. V tuto chvíli se celý proces opakuje jako dříve mezi Alicí a prvním směrovačem. Po dojednání a ustanovení okruhu OR 1 zašle potvrzovací buňku s příkazem rozšířeno a zašifrovanými informacemi pro dohodnutí klíče $k_2=g^{x2y2}$, tím Alice a OR 2 sdílí privátní klíč. Celý tento proces může pokračovat a mohou se tak vytvářet další cesty dle potřeby. V základu jsou použity tři cibulové směrovače při komunikaci ven ze sítě Tor.

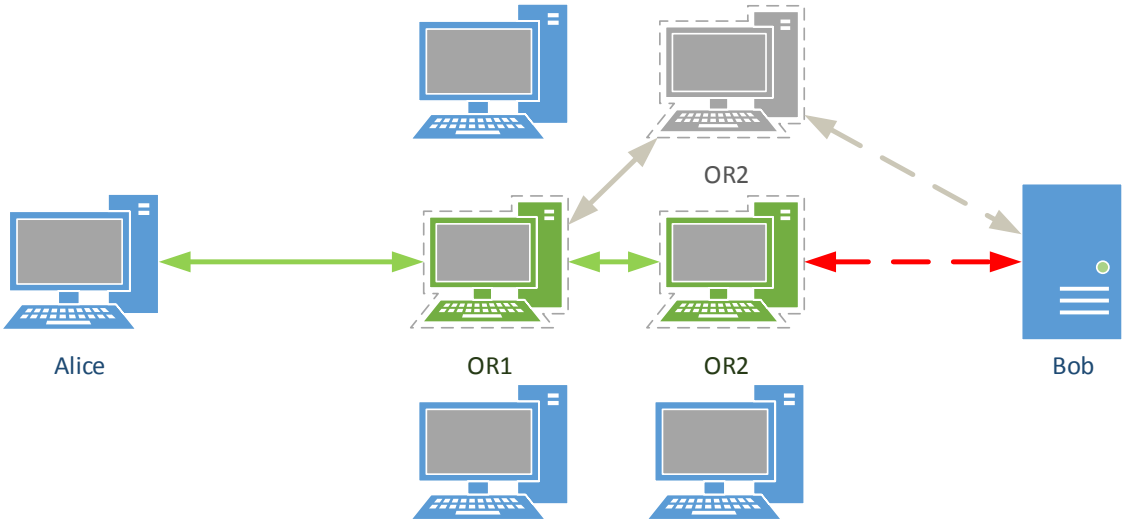
Výsledkem těchto výměn je, že Alice zná všechny klíče potřebné pro komunikaci. Jednotlivé směrovače znají pouze klíč svého souseda a zároveň ani jeden z těchto směrovačů není schopen určit kolikátý je v pořadí. Alice spolu s tím, že nebyl zveřejněn žádný její veřejný klíč, zůstává v anonymitě.

Alice má již sestavenou kompletní síť okruhů nutnou ke komunikaci. Může tedy začít posílat okruhové buňky s požadavky například na webový server. Vyšle tedy zprávu směrem k OR 1, která je zašifrována všemi klíči, které budou potřebné a obsahující všechny hlavičky.

Podoba této zprávy připomíná složení vrstev cibule, z tohoto důvodu bylo toto směrování pojmenováno Cibulové směrování. OR 1 po příchodu zprávy porovná, zda číslo okruhu koresponduje s jeho čísly a pokud ano, rozbalí zprávu a následně ji zašle dál směrem k OR 2, tím odstraní pomyslnou cibulovou slupku, jež dala jméno tomuto směrování.

Zpětná komunikace od cílové stanice směrem k Alici je podobná. OR 2 pošle buňku zpět OR 1 a zabezpečí ji jejich sdíleným klíčem. Bob dešifruje zprávu a určí, jestli je platná. Následně provede zabalení do další vrstvy, kterou zabezpečí klíčem, který sdílí s Alicí a přidá její adresu.

OP Alice má také možnost rušit sestavené okruhy. Alice může zaslat zprávu „relay truncate“, tím dojde k ukončení spojení v definovaném okruhu, ale není tím poškozeno celé spojení, které bylo vytvořeno. Alice tedy může zrušit spojení mezi OR 1 a OR 2, ale OR 1 mezi tím může navázat spojení s jiným směrovačem a začít komunikovat.




**Obr. 2.8. Rušení části okruhového spojení**

Stejně jednoduchým a rychlým způsobem je možné zrušit celé vytvořené spojení. Alice vyšle zprávu „znič“, která dojde OR 1. Ten okamžitě zruší své spojení s Alicí, v podobě uzavření kanálů a okruhu, který měl doposud s Alicí sestaven, zašle tuto zprávu dalším OR, které provedou to samé. To má za následek velmi rychlé ukončení spojení, protože Alice nečeká, až se zruší nejvzdálenější spojení, ale ukončí spojení, které je jí nejblíže a o zbytek se již nemusí starat.

### Kontrola zahlcení

Síť Tor, jak již bylo zmíněno, je tvořena dobrovolníky, kteří nejsou ochotni sdílet celé své síťové prostředky a odrážet tak výkyvy linky. Uživatelé raději poskytnou pevnou část linky, která je samotné nijak neomezí. Tento fakt může být zneužit a použit jako jeden z možných útoku na síť Tor. Jestliže útočník bude ovládat cílový server, a on sám bude vytvářet dostatečně velký datový provoz, který cílový server řízený útočníkem odmítne přijmout, dojde k zahlcení OR a tím i dalších uživatelů. To by při správném použití mohlo poškodit všechny uživatele a především celou síť Tor.

Z těchto důvodu bylo nutné použít nějaký druh řízení linky. Za vhodné řešení bylo zvolen princip token bucket [15], který hlídá množství přijatých a odeslaných buněk. OR využívá dvou oken. První okno zvané balící sleduje množství buněk, které je OR ochoten přijmout. Druhé okno zvané doručovací sleduje množství buněk, které je OR ochoten odeslat mimo síť. Obě tato okna dekrementují hodnotu uloženou v kyblíku. Pokud OR obdržel dostatek buněk, odešle okruhovou zprávu „relay sendme“ se „streamID“ nastaveným na nulu směrem k OP. Jakmile přilehlý OR přijme tuto zprávu inkrementuje hodnotu balícího okna, pakliže tomu tak není, dojde k vyčerpání kyblíku a OR již nepřijme další data. Obdobně se chová i OP, který jako řídící prvek celého spojení musí kontrolovat jednotlivá okna u všech OR.

Dalším způsobem potlačení zahlcení je řízení mezi koncovými zařízeními, kdy se nastaví určitá hodnota kyblíku pro jednotlivé datové proudy a OP pravidelně zasílá okruhové zprávy „sendme“, které inkrementují hodnotu kyblíku o určitou hodnotu. Zároveň je kontrolováno, zdali jsou data úspěšně odesílána. Pokud jsou a dojde k poklesu hodnoty v kyblíku pod určitou hodnotu, teprve tehdy je odeslána další buňka „sendme“.

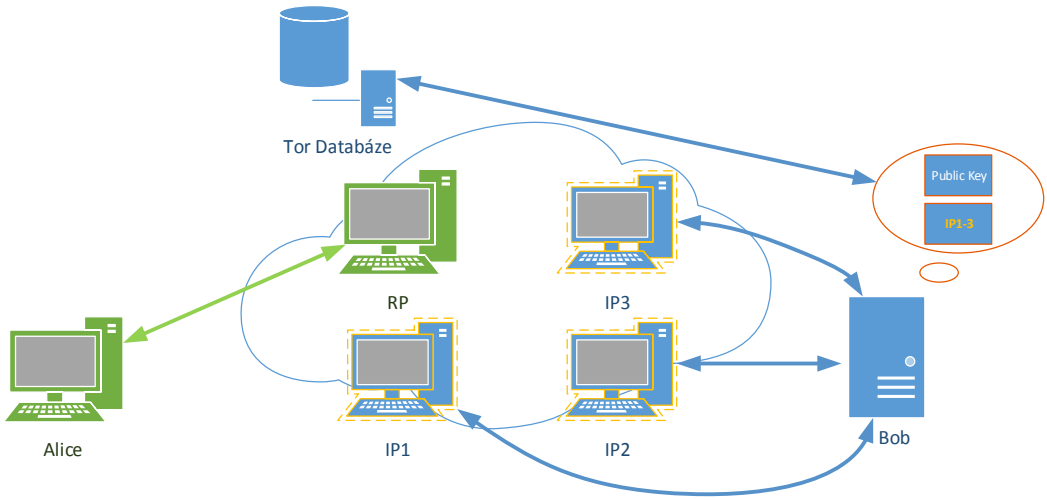
## 2.3.4.Skrytá služba (Hidden service)

Tor kromě anonymního prohlížení internetu nabízí i mnoho jiných služeb, které mohou být použity k poskytování anonymních informací. Zjednodušeně si to lze představit jako intranet sítě Tor. Je zde možno provozovat například vlastní webový server, chat, SSH server a mnohé jiné. To vše lze využít k výměně informací mezi poskytovatelem a návštěvníkem, aniž by byla prozrazena jejich totožnost.

Cílem těchto Skrytých služeb je, aby nebyla vyzrazena IP adresa serveru (Boba), zamezení DoS útokům na jeho službu. Dále nutnost udržet dlouhodobě skrytou identitu pro navázání spojení s Bobem a to nezávisle na několika OR. Umožnění Bobovi odmítnutí komunikace s Alicí a v neposlední řadě nutnost jednoduché implementace Skryté služby.

## Postup sestavení a navázání skryté služby:

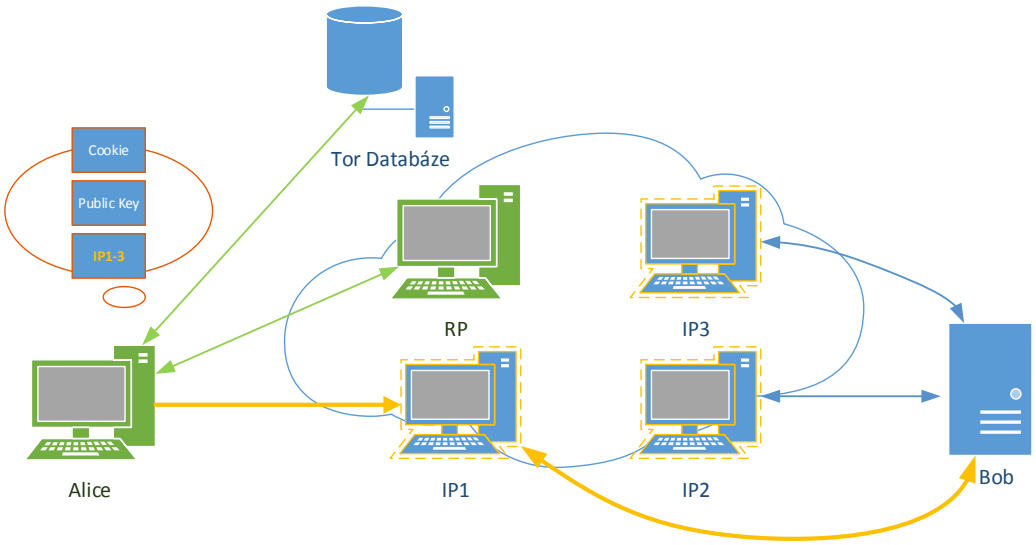
1. V prvním kroku je nutné vygenerovat veřejný a privátní klíč. Dále oznámit existenci Bobova serveru. Z tohoto důvodu Bob náhodně kontaktuje několik uzlů, které budou sloužit jako body pro seznámení (Introduction Point – IP), vybuduje si k nim okruhová spojení a sdělí jim svůj veřejný klíč.
2. Následně Bob informuje vyhledávací službu (lookup service) a oznámí jí popis Skryté služby, který obsahuje jeho veřejný klíč a informace o kontaktních uzlech. To vše je podepsáno Bobovým privátním klíčem a odesláno. Zároveň je z Bobova veřejného klíče vygenerován šestnáctisymbolový popis, ve formátu XXX.onion, za pomoci kterého bude Bobova služba dostupná. Tento identifikátor je taktéž zaslán do veřejné hašovací databáze. Kontaktní uzly jsou zároveň informovány, aby začaly naslouchat požadavkům Bobovy služby.




**Obr. 2.9. Počáteční sestavení okruhů pro skrytou službu**

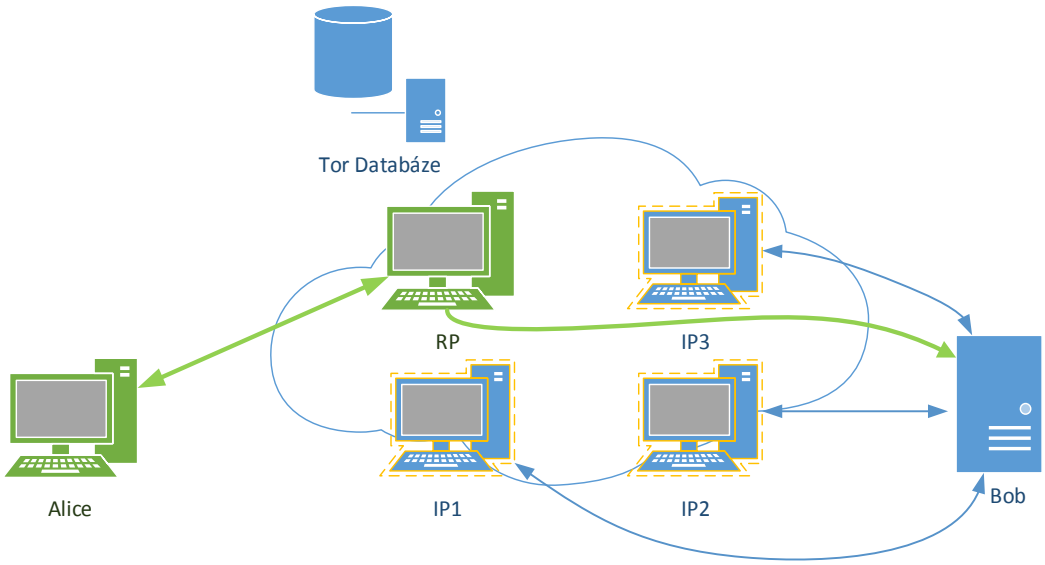
3. Serverová část spojení je nyní připravena přijímat požadavky. Alice nějakým způsobem, z internetu, zjistila adresu Bobovy služby na adrese XXX.onion. Jestliže se chce připojit k Bobovi anonymně, využije sítě Tor ke kontaktování vyhledávací služby a dotáže se na XXX.onion. Vyhledávací služba následně poskytne Alici veřejný klíč Boba (pokud existuje) a soubor přístupových bodů, přes které je dostupný.

4. Alice si zvolí jeden náhodný OR jako rendezvous bod (RP), který bude použit ke komunikaci s Bobem, vybuduje si k němu spojení a předá mu rendezvous cookie, která obsahuje jednorázové tajemství.
5. Následně Alice sestaví seznamovací zprávu, která obsahuje informace o Alici, její RP, začátek Diffie-Hellmanovy výměny klíčů a rendezvous cookie. To vše je podepsáno Bobovým veřejným klíčem, takže pouze on může zprávu dešifrovat. Vše je zasláno na jeden seznamovací bod (IPx), který Alice obdržela od vyhledávací služby. Příchozí zprávu následně OR přepošle Bobovi a ten se rozhodne, zdali bude s Alicí komunikovat.




**Obr. 2.10. Navázání spojení skrze Bobovy body pro seznámení (IPx)**

6. Má-li Bob má zájem o komunikaci s Alicí, vybuduje si připojení na domluvený RP a zašle na něj druhou půlku Diffie-Hellmanovy výměny klíče, rendezvous cookie a haš klíče spojení, které právě používají, obdobně jako při navazování spojení mezi OP a OR. Obě strany nyní mají vyměněny klíče a vědí navzájem o sobě. Rendezvous bod následně spojí dvě tato spojení dohromady, netušící nic o Alici, Bobovi a datech která si vyměňují.
7. Alice zašle zprávu „relay begin“, která informuje o začátku komunikace a zašle ji skrze celou cestu k Bobovu OP, který zaručí dodání Bobově běžící službě.




**Obr. 2.11. Navázané spojení mezi uživatelem a skrytou službou**

### 2.3.5.Útoky na síť Tor

Anonymní komunikace v dnešní době je brána na lehkou váhu. Lidé o sobě především na Internetu sdělují velké množství informací a to vědomě či nevědomě. Získání a následné použití těchto dat je extrémně ceněnou komoditou, ať už se jedná o cílenou reklamu, různá doporučení až po sledování lidí korporacemi nebo vládami. Z dnešního pohledu je nežádoucí, když se někdo snaží skrýt svoji identitu, a tím na sebe paradoxně upozorňuje. Stejně je tomu i se sítí Tor, která tím, že poskytuje zdarma a všem anonymní pohyb, se vystavuje velkému přílivu útoku, snažících se zdiskreditovat celou síť.

Mnohé zde zmíněné útoky jsou všeobecně známé a dostupné, což umožňuje návrhy útoků, tak i následné implementace oprav. Druhá skupina útoků není veřejně dostupná a je těžší se proti ní bráni, avšak k těmto slabým místům má přístup pouze omezený počet osob.

Jedním z největších útočníků, je paradoxně sama americká vláda, která sama financuje chod Toru. Je nutné upozornit, že zde existuje nepřeberné množství vlád, korporací a jednotlivců snažících se o účinný útok vůči této síti a jejich uživatelům. Získání informací o těchto pokusech je téměř nemožné, pokud nejsou uveřejněny. K tomuto došlo v kauze bývalého zaměstnance Národní bezpečností agentury (NSA) Edwarda Snowdena, který v červnu 2013 předal tajné informace a o sledování osob na Internetu [26]. Mezi těmito informacemi se vyskytují prezentace a dokumenty popisující způsoby útoků americké vlády na síť Tor. I přes neověřitelnost těchto informací, budou některé druhy útoků pocházejících z těchto materiálů zmíněny, jelikož popisují způsoby, kterými by se mělo dát zaútočit na síť Tor.

## Odmítnutí služby:

Odmítnutí služby (Denial of service – DoS) je jednou z nejznámějších možností útoku silou na cílovou stanici, při které útočník zahlcuje cílovou stanici nesmyslnými požadavky, které není schopna zpracovat. Tím dochází k nedostupnosti služby. Tento útok je také často distribuovaný (DDoS), kdy velké množství požadavků je generováno velkým množstvím útočníků, kteří jsou zaměřeni na jednu cílovou službu. Nutností pro tento útok je znalost cílové stanice, to je však v síti Tor problematické, jelikož útočník nezná identitu a adresu klienta, na kterého chce zaútočit. V případě skryté služby Tor, je možný nepřímý útok vůči servu a to prostřednictvím seznamovacích bodů, které jsou použity jako kontaktní body pro přístup na server.

Částečným řešením je zvětšení počtů těchto přístupových bodů a tím nutnosti zaútočit na všechny. Další obranu proti tomuto útoku může poskytnout řízení zahlcení linky, které omezí počty útoků, omezením přijatých zpráv. Stejným způsobem je možné zaútočit vůči OR a zahlcovat jej požadavky na ustanovení TLS spojení, které při výměně klíčů velmi vytěžuje linku i samotný výpočetní výkon OR. Řešením této situace v podobě řízení zahlcení je nežádoucí, protože služba je ochráněna, ale je omezen v mnoha případech i zamezen přístup ostatních uživatelů, což může být i jeden z cílů útočníka.

Řešení, které by pomohlo vyřešit tuto situaci je aktivita samotného uživatele, který se snaží přistupovat ke službě, jelikož DoS útoky jsou jednosměrné a nevyžadují interakci uživatele či útočníka. Pomoci s tímto problémem může Client Puzzle Protocol (CPP - Klientský puzzlový protokol). Ten v případě požadavku na TCP spojení zašle klientovi jednoduchý obrázek, který je nutné matematicky vyřešit a zaslat zpět. Server rychle přijme spojení nebo ho zahodí. Toto řešení tak nepředstavuje extrémně velké zatížení, stejně jako řešení CAPTCHA kódů. Ve výsledku se tedy jedná o jednoduchý Turingův test na odhalení skutečného uživatele od umělé inteligence.

## Rejstříkové servery:

Dalším z možných útoků je zneužití rejstříkových serverů (DS - Directory Servers). V prvním návrhu cibulového protokolu byl princip výměny informací o OR dosti podobný protokolu OSPF, kde jsou všichni sousedé zahlceni zprávami o nově připojeném uzlu. To společně s účelným či náhodným zpožděním buněk způsobovalo nejednotný pohled na topologii sítě a různé přístupy ke směrování. Navíc rostoucí počet uživatelů a expanze sítě umožňoval útočníkovi podvrhnout falešné směrovače či části celé sítě.

Z tohoto důvodu bylo upuštěno od tohoto směrování a místo něho je implementován systém několika redundantních cibulových směrovačů, které monitorují, ověřují chování, topologii sítě Tor, včetně klíčů a výstupních bodů. Takto získané informace jsou zaslány na rejstříkový server, který se pro uživatele tváří jako klasický HTTP server. Z něj mohou dle potřeby stáhnout přehled o topologii sítě pro navázání spojení s OR, stejně tak OR informují rejstříkové servery o svém aktuálním stavu a dostupnosti. Klientská aplikace zároveň obsahuje seznam DS a jejich klíče, aby bylo možné po spuštění klienta aktualizovat topologii.

Po příchodu informací od OR na DS je nutné ověřit, zdali klíče od OR jsou známé, jelikož nesmí dojít k zahrnutí OR, který je nerozpoznaný, do celé databáze. To by mohlo mít za následek vytvoření mnoha serverů a ovládnutí celé sítě. Do průběhu schvalování zasahuje také DS administrátor, který potvrzuje jednotlivé nové OR. To má může být zároveň výhodou i jednou ze slabin schvalovacího procesu.

Útoky na tento server jsou nasnadě. V případě získání nadvlády nad tímto serverem umožníme útočníkovi podvrhnout různé informace, které mohou být použity ke zjišťování chování uživatele nebo ke směrování do útočníkem ovládané sítě.

V aktuální době existují tři rejstříkové servery, kdy je plánován rozvoj s rostoucí popularitou Toru. Je nutné, aby tyto servery byly mezi sebou vždy aktualizované a byly schopny se dohodnout na jednotné databázi. Uživatel zároveň musí věřit pouze těmto serverům a nikomu jinému. DS jsou navrženy tak, aby pracovaly co nejjednodušeji. Zároveň spoléhají v případě problémů s databázemi na lidský zásah, který pomůže vyřešit tento problém.

V případě, že se OR dostane do všech databází a odmítne následně přeposílat datový provoz, mělo by to za následek výpadek části sítě. Řešením by mohlo být průběžné anonymní testování jednotlivých OR na jejich propustnost.

Útočník v případě zničení či zablokování DS neznemožní komunikaci v síti Tor, jelikož jsou všechny databáze zrcadlené, avšak získání nadpolovičního množství DS by vedlo ke kompromitaci celé topologie díky synchronizaci, která by změny zkopírovala na zbylé databáze. Obrana před tímto útokem opět spočívá na lidském faktoru, který by měl zamezit těmto pokusům.

## Pasivní útoky:

Dalším z možných útoků vůči síti Tor představují pasivní útoky, které nijak nepoškozují síť či uživatele, ale snaží se determinovat uživatele podle jeho chování. Tento způsob nevyžaduje velký výpočetní výkon a časovou náročnost. Vyžaduje však přístup k podezřelému uživateli, kterého musíme již znát a přístupům k datovému toku velkého počtu serverů.

Jak již bylo zmíněno, datová komunikace od výchozího budu, v podobě Alice, k prvnímu OR v podobě Boba je již šifrována a tudíž zde není možnost, v přiměřené časové době, prolomit toto šifrování. Je tedy nutné provádět datovou analýzu procházejících paketů a tu následně porovnávat s datovým tokem na cílovém serveru. V případě objevení těchto vzorců komunikace, je možné využít datového toku u cílové stanice k přečtení obsahu, jelikož komunikace mezi výstupním směrovačem Tor a cílovou stanicí nemusí být šifrována, a tím může poskytnout potřebné informace o Alici a jejích datech. Tento způsob sledování může být zdokonalen vybudováním databáze, která bude obsahovat otisky, v podobě množství a časových přístupů, pro jednotlivé cílové servery. Následně stačí, aby útočník porovnal chování zkoumaného počítače vůči zaznamenané databázi.

Tento způsob se dá také využít k rozlišení jednotlivců a skrytých služeb, kde se k získaným informacím z databáze přidají cibulové adresy serverů a následně se determinuje zdroj.

## Aktivní útoky:

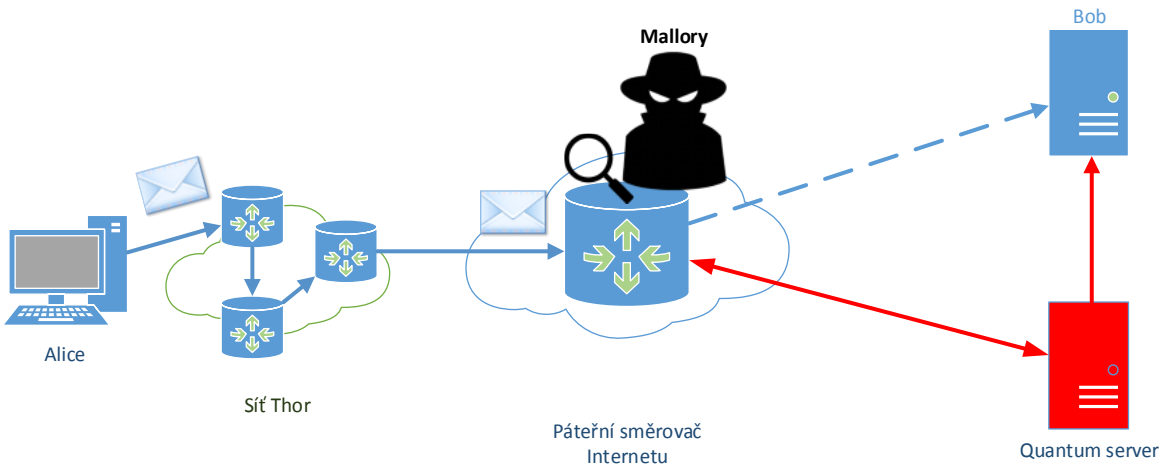
Aktivní útoky, na rozdíl od pasivních útoků, vyžadují velkou míru znalosti sítě a velmi velkých nároků na techniku a její výpočetní výkon. Útočník v těchto případech aktivně zasahuje do celé komunikace a snaží se zdiskreditovat jednotlivé účastníky komunikace.

Jeden z těchto útoků může být veden vůči cibulovým směrovačům, za pomoci nichž se bude snažit útočník skok po skoku dostat až k původci celé komunikace. Problém tohoto reverzního postupu spočívá v tom, že OP hlídá životnost jednotlivých toků a v pravidelných časových blocích tato spojení ruší. Útočník tedy není schopen dostatečně rychle determinovat všechny body komunikace. Zároveň získání nadvlády nad OR neumožňuje dešifrování komunikace, protože veškeré informace jsou již zničeny díky uzavření okruhu.

Dalším z těchto útoků je útok na TLS klíč OR, kterým jsou podepsána data mezi OP a OR. Prolomením privátního klíče cibulového směrovače by bylo možné zjistit a ovládat zaslané zprávy a to v rámci všech sestavených okruhů, díky klíči jednotlivých relací. Problém tohoto útoku spočívá v nutnosti znát cibulový klíč, aby bylo možné ovládnout zprávy určené k sestavování jednotlivých okruhů. To však kvůli dopředné výměně klíčů není možné. Útočník tedy není schopen ovlivnit a převzít řízení nad již existujícím kruhovým spojením, jelikož by zároveň ovlivnil i klíče pro jednotlivé relace. Navíc další vrstvou ochrany je vytváření nových klíčů v pravidelných cyklech, což znemožňuje rychlé dešifrování klíče a jeho uplatnění.

Jedním z dalších útoků může být porovnávání dostupnosti klienta. Pokud bude uživatel používat svůj počítač k přístupu k Internetu přes veřejné spojení, a zároveň jej využívat k běhu Skryté služby v rámci Toru, je možné determinovat jednoduchým dotazem či DoS útokem na server jeho příslušnost ke Skryté službě. Útočníkovi pak stačí ověřovat dostupnost uživatelské stanice z veřejné sítě a zároveň ze sítě Tor. V případě nedostupnosti je jasné, že na uživatelském počítači běží Skrytá služba sítě Tor. Obrana proti tomuto útoku spočívá v zamezení ICMP zpráv, které slouží k ověření dostupnosti. Dále vytvořením virtuálního serveru hostujícího Skrytou službu. Ten v případě poruchy či napadení neprozradí lokaci, protože fyzické zařízení je stále funkční a dostupné z Internetu.

V materiálech NSA je také popsán útok využívající jeden z nejznámějších problému kryptografie a informatiky. Tím je útok člověka uprostřed (Man in the Middle), kdy se do komunikace mezi Alicí a Bobem aktivně zapojí třetí osoba tzv. Mallory. Tento princip posloužil v případě NSA k vytvoření tajných serverů známé pod názvem Quantum. Ty jsou umístěny na páteřních sítích s nutnou spoluprací internetových poskytovatelů a velkých společností jako Google, Microsoft a jiné. Díky takto vhodnému umístění, jsou servery schopny odpovědět rychleji na nezabezpečený požadavek „HTTP get“ než samotný cílový server.

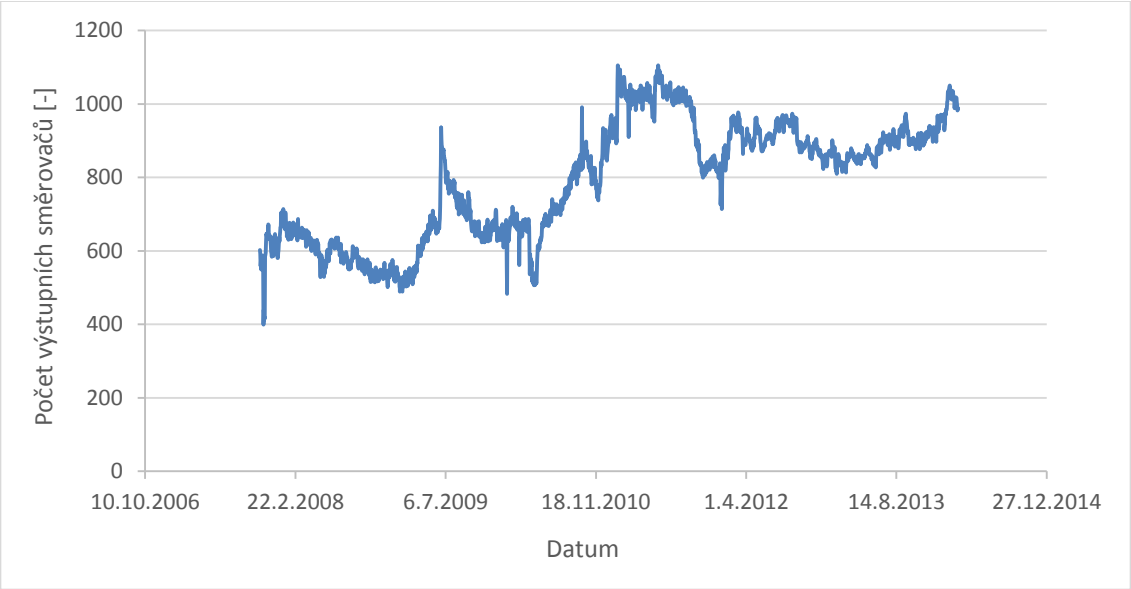



**Obr. 2.12. Útok typu Člověk uprostřed**

Takto podvržené zprávy jsou využívány k přesměrování na další servery pod nadvládou NSA s názvem FoxAcid server [19]. Tyto servery jsou běžně dostupné servery, na kterých běží upravený program, který čeká na příchod požadavků se speciální adresou tzv. FoxAcid značkou (FoxAcid tag). Na přesměrované zařízení je následně zaútočeno a to různými druhy útoků v závislosti na důležitosti cíle.

## a) Zkažené cibule

Nejzranitelnějším místem sítě Tor jsou jeho výstupní směrovače, přes které je směrován všechen datový provoz určený do „nezabezpečeného“ Internetu. Těchto směrovačů je přibližně kolem jednoho tisíce, přes které je směrován datový provoz o velikosti řádově v jednotkách GiB/s [25]. Tyto směrovače se chovají jako MitM, kdy vstupují do komunikace mezi uživatelem a cílovou stanicí. To vše předurčuje tyto výstupní body k cíleným útokům.




**Graf 2.1. Závislost počtu výstupních směrovačů v čase**

V práci [28] byl vytvořen program ExitMap, který má za cíl odhalit možné útoky, kterých se může dopustit hraniční směrovač pod kontrolou útočníků. Tento program byl navržen tak, aby testoval jednotlivé hraniční směrovače a porovnával reálné a obdržené hodnoty.

Z výsledků měření byly pozorovány tři hlavní způsoby útoků na datovou komunikaci.

### b) HTTPS MitM útok

Nejčastěji využívaným protokolem v síti Tor je stále nezabezpečený protokol HTTP, neposkytující žádné zabezpečení, následován jeho zabezpečenou verzí HTTPS. Tento protokol je, i přes své zabezpečení, cílem útoků MitM a to v podobě pokusů podvrhnout certifikát serveru při pokusu o sestavení spojení a to vlastnoručně podepsaným certifikátem.

Tyto útoky, nejsou prováděny plošně na veškerou komunikaci, ale jsou aplikovány pouze na každé n-té spojení se zaměřením na nejpopulárnější stránky jako facebook.com. Úspěch těchto serverů není příliš vysoký, jelikož při použití podvrženého certifikátu prohlížeč oznámí tuto skutečnost uživateli, protože certifikát nebyl ověřen důvěryhodnou CA a je tedy nedůvěryhodný.

### c) SSH MitM útok

Obdobně jako v předcházejícím případě může být síť Tor použita k přenosu datového provozu skrze SSH. Pokus o MitM útok je zde mnohem těžší oproti útoku na HTTPS, protože při úspěšném sestavení spojení je veřejný klíč serveru uložen a použit k následnému porovnání při dalším spojení. Nutností úspěchu tohoto útoku je tedy zachycení první zprávy při pokusu o navázání spojení, to je velmi obtížné a nahodilé.

### d) SSLstrip útok

Dalším rozšířeným útokem je SSLstrip, který funguje opět na principu MitM a který zamezuje sestavení TLS spojení. Princip tohoto útoku je v nahrazení zabezpečeného protokolu HTTPS za protokol HTTP a tím použití nezabezpečeného spojení. I v dnešní době je možné přistupovat na různé webové stránky oběma protokoly a změnou na protokol HTTP je možné následně odposlechnout datovou komunikaci včetně nekódovaných hesel.

### e) DNS útok

Síť Tor využívá v případě dotazů na DNS spojení TCP, kde se dotaz klienta zabalí do zprávy Relay Begin. Tato zpráva je následně na hraničním směrovači rozbalena a zaslána k přeložení příslušnému DNS serveru.

Tímto způsobem je možné ovládat a podvrhovat jednotlivé DNS záznamy. Tento útok nemusí být vždy útokem. Příkladem může být turecký výstupní směrovač, který je blokován lokální vládou tak, aby přesměroval stránky s pornografickým obsahem na stránky tamní vlády. Obdobný způsob byl identifikován i na směrovači v Číně.

Během čtyř měsíců bylo několikrát týdně testováno přibližně 950 výstupních směrovačů, ze kterých se ukázalo 25 jako potencionálně nebezpečných nebo špatně nakonfigurovaných. Výsledky měření jsou shrnuty níže v tabulce.

**Tab. 2.1. Zkažených cibulových směrovačů a druhy útoků**

| Otisk | IP adresa | Země | Kapacita linky | Útok | Četnost výskytu | Aktivní od | Objeveno |
|---|---|---|---|---|---|---|---|
| **F8FD29D0 †** | 176.99.12.246 | Rusko | 7,16 MB/s | HTTPS MitM | neznámá | 24.6. 2013 | 13.7. 2013 |
| **8F9121BF †** | 64.22.111.168/29 | USA | 7,16 MB/s | HTTPS MitM | neznámá | 11.6. 2013 | 13.7. 2013 |
| **93213A1F †** | 176.99.9.114 | Rusko | 290,00 KB/s | HTTPS MitM | 50,00% | 23.7. 2013 | 19.9. 2013 |
| **05AD06E2 †** | 92.63.102.68 | Rusko | 5,55 MB/s | HTTPS MitM | 33,00% | 1.8. 2013 | 19.9. 2013 |
| **45C55E46 †** | 46.254.19.140 | Rusko | 1,54 MB/s | SSH + HTTPS MitM | 12,00% | 9.8. 2013 | 23.9. 2013 |
| **CA1BA219 †** | 176.99.9.111 | Rusko | 334,00 KB/s | HTTPS MitM | 37.50% | 26.9. 2013 | 1.10. 2013 |
| **1D70CDED †** | 46.38.50.54 | Rusko | 929,00 KB/s | HTTPS MitM | 50,00% | 27.9. 2013 | 14.10. 2013 |
| **EE215500 †** | 31.41.45.235 | Rusko | 2,96 MB/s | HTTPS MitM | 50,00% | 26.9. 2013 | 15.10. 2013 |
| **12459837 †** | 195.2.252.117 | Rusko | 3,45 MB/s | HTTPS MitM | 26.90% | 26.9. 2013 | 16.10. 2013 |
| **B5906553 †** | 83.172.8.4 | Rusko | 850,90 KB/s | HTTPS MitM | 68,00% | 12.8. 2013 | 16.10. 2013 |
| **EFF1D805 †** | 188.120.228.103 | Rusko | 287,60 KB/s | HTTPS MitM | 61.20% | 23.10. 2013 | 23.10. 2013 |
| **229C3722** | 121.54.175.51 | Hong Kong | 106,40 KB/s | SSL strip | neměřeno | 5.6. 2013 | 31.10. 2013 |
| **4E8401D7 †** | 176.99.11.182 | Rusko | 1,54 MB/s | HTTPS MitM | 79.60% | 8.11. 2013 | 9.11. 2013 |
| **27FB6BB0 †** | 195.2.253.159 | Rusko | 721,00 KB/s | HTTPS MitM | 43.80% | 8.11. 2013 | 9.11. 2013 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **0ABB31BD †** | 195.88.208.137 | Rusko | 2,30 MB/s | SSH + HTTPS MitM | 85.70% | 31.10. 2013 | 21.11. 2013 |
| **CADA00B9 †** | 5.63.154.230 | Rusko | 187,62 KB/s | HTTPS MitM | neměřeno | 26.11. 2013 | 26.11. 2013 |
| **C1C0EDAD †** | 93.170.130.194 | Rusko | 838,54 KB/s | HTTPS MitM | neměřeno | 26.11. 2013 | 27.11. 2013 |
| **5A2A51D4** | 111.240.0.0/12 | Taiwan | 192,54 KB/s | HTTP Injekce | neměřeno | 23.11. 2013 | 27.11. 2013 |
| **EBF7172E †** | 37.143.11.220 | Rusko | 4,34 MB/s | SSH MitM | neměřeno | 15.11. 2013 | 27.11. 2013 |
| **68E682DF †** | 46.17.46.108 | Rusko | 60,21 KB/s | SSH + HTTPS MitM | neměřeno | 2.12. 2013 | 2.12. 2013 |
| **533FDE2F †** | 62.109.22.20 | Rusko | 896,42 KB/s | SSH + HTTPS MitM | 42.10% | 6.12. 2013 | 8.12. 2013 |
| **E455A115** | 89.128.56.73 | Španělsko | 54,27 KB/s | SSLstrip | neměřeno | 18.12 2013 | 18.12 2013 |
| **02013F48** | 117.18.118.136 | Hong Kong | 538,45 KB/s | DNS cenzura | neměřeno | 22.12. 2013 | 1.1. 2014 |
| **2F5B07B2** | 178.211.39 | Turecko | 204,80 KB/s | DNS cenzura | neměřeno | 28.12. 2013 | 6.1. 2014 |
| **4E2692FE** | 24.84.118.132 | Kanada | 52,22 KB/s | Open DNS | neměřeno | 21.12. 2013 | 6.1. 2014 |

Jak je patrné, většina zkažených směrovačů pochází z Ruska a podle podpisů certifikátu [28] je taktéž patrné, že směrovače označené křížkem patří jednomu útočníkovi. Většina provedených útoků je vedena s cílem podvrhnout certifikát uživateli a spoléhá na jeho důvěru a neznalost.

### *f)* Zneužití uživatele:

Mnohé útoky je možné vést vůči samotnému uživateli, který se svojí lehkovážností a neopatrností může sám zdiskreditovat a to následovně:

Stažením aplikace z neznámého zdroje, na kterém je modifikovaná aplikace. Projekt Tor vydává aplikaci vždy podepsanou, je tedy nutné ověřit původ a správnost této aplikace před nainstalováním a prvním použitím. Aplikace může dělat pravý opak anonymizace a bude se snažit zjistit a odeslat co nejvíce informací o uživateli. Projekt Tor kromě podpisu a umístění aplikace na důvěryhodné servery také poskytuje otevřenost

zdrojových kódů, takže jsou všichni schopni ověřit chování jednotlivých součástí aplikace.

Dalším způsobem je využití sledovacích cookie souborů, které webové servery vytváří k identifikaci a personalizaci jejich webových stránek. Ty používají společnosti jako Google k šíření personalizované reklamy. Jsou zaznamenány případy, kdy tyto soubory přečkaly v anonymním prohlížeči Toru, který běžel na upravené verzi prohlížeče Firefox. Tato bezpečnostní chyba se vyskytovala v prohlížeči Firefox, ve verzích 10 ESR a 11.0-16.0.2, který využíval knihovnu E4X. Aktuální možnosti zneužití prohlížeče nejsou veřejně dostupné, pokud však uživatel bude dodržovat základní doporučení, mělo by být velmi obtížné penetrovat klientské zařízení, jelikož defaultně je zakázán Flash, skripty a zároveň je uživatel upozorněn, aby nepoužíval různá rozšíření prohlížeče.

Nejúčinnější ochranou je tak použití separátních prohlížečů pro běžnou a zabezpečenou komunikaci, nejlépe s odděleným operačním systémem určeným pouze pro účely anonymního pohybu.

### 2.3.6. Útoky na Skrytou službu

#### Útok na adresáře Skrytých služeb

V práci [2] publikované na bezpečnostním sympoziu o bezpečnosti 2013 byl publikován způsob, jakým je možné zjistit popularitu a statistiky přístupů na jakoukoliv Skrytou službu sítě Tor a to prostřednictvím databázových adresářů (HSDir) [17].

Uživatel přistupující na Skrytou službu prvně musí získat informace o IP, které slouží pro navázání spojení. K tomu využije některou z mnoha HSDir, které evidují informace pro kontaktování Skryté služby. K tomu, aby cibulový směrovač obdržel vlajku HSDir, je pouze nutné splnit podmínku minimální doby funkčnosti 25 hodin. Cílem útočníka je tedy provozovat databáze a být zvolen Skrytou službou jako uložiště přístupových údajů. S těmito informacemi je možné na zkompromitovaném směrovači sbírat statistiky přístupů a hlavně zabránit v přístupu na server podáváním nepravdivých nebo žádných údajů.

Skrytá služba informace o své službě replikuje do celkem šesti databází s cílem dosáhnout co nejlepšího pokrytí v případě výpadku jednoho ze směrovačů. Toho dosáhne vypočtením descriptor-id následovně [16]:

```
descriptor-id = H(permanent-id | H(time-period |          (2.1)
 | descriptor-cookie | replica))
```

Tato hodnota je vypočtena pomocí permanent-id, která je složena z veřejného klíče (adresa serveru), dále pak descriptor-cookie, což je volitelné číslo, ve většině případů prázdné. Následováno replikou, která je složena z čísla integer. Ta rozdělí identifikátor na sadu dvou identifikátorů po třech HSDir. Poslední proměnou je time-

period, která se mění každých 24 hodin. Výpočet této proměnné je proveden z aktuálního času a bytu permanent-id z důvodu, aby se všechny Skryté služby neaktualizovali ve stejný čas.

```
time-period = (current-time + permanent-id-byte *        (2.2)
* 86400 / 256) / 86400
```

Skrytá služba následně nalezne tři databázové adresáře s identifikátorem větším než právě vypočítaný a předá jim své informace. Identifikátor je zván identity-digest a je vypočítán z SHA1 haše veřejného klíče, který přidělila síť Tor směrovači.

```
identity-digest = H(server-identity-key)                 (2.3)
```

Z výše zmíněnými proměnnými může útočník předem vypočítat deskriptor Skryté služby a identifikátory HSDir pro specifikované časové období. Následně stačí, aby útočník měl v provozu minimálně 25 hodin cibulové směrovače s vypočtenými identifikátory a vyčkal, až jej HS kontaktuje.

Tento útok není nijak nákladný na prostředky. Je za potřebí minimálně 18 směrovačů fungujících jako databázové adresáře, aby byla zaručena kontinuální blokace. Množství požadovaných IP adres může být také radikálně snížen a to využitím slabiny návrhu Toru.

Tor umožňuje na jedné IP adrese maximální souběh dvou aktivních směrovačů, avšak monitoruje vlastnosti i neaktivních směrovačů na téže IP adrese. V případě vypnutí jednoho z aktivních směrovačů jeho roli převezme jeden z pasivních směrovačů, který nemusí čekat 25 hodin na přidělení vlajky HSDir.

Možnou ochranou je, aby Skrytá služba periodicky kontrolovala svoji dostupnost z HSDir a v případě nedostupnosti kontaktovala správce Toru. Dále je vhodné použít málo využívaného descriptor-cookie, které slouží jako tajné heslo, ale zároveň zvětšuje délku adresy z 16 znaků na 42. Posledním opatřením je zamezení sbíraní informací o pasivních směrovačích, které nemohou okamžitě po aktivaci obdržet status HSDir.

*Autentizace při přístupu na HS*

V předchozím útoku bylo zneužito HSDir k útoku na IP, které slouží k navázání spojení se Skrytou službou. Jednou z navrhovaných ochran je použití zmíněného descriptor-cookie k zašifrování informací o IP.

V návrhu Toru je implementována ochrana autorizací, která umožňuje omezit okruh přistupujících uživatelů k serveru hostujícím Skrytou službu [13].

Tato ochrana se dělí na dva autorizační protokoly:

*Základní „basic“ autorizace*

Cílem tohoto protokolu je poskytnout službu co největšímu počtu uživatelů bez velkého vlivu na výkon. Protokol spoléhá na znalost symetrického klíče, v podobě descriptor cookie, který je použit pro zašifrování informací v descriptor-id. Díky tomu je zamezeno neoprávněnému přístupu na server, jelikož uživatel se znalostí adresy služby nebude schopen získat informace o IP a tím se připojit. Přístup je tedy umožněn pouze uživatelům, kteří nějakou cestou obdržely tento klíč.

Nevýhodou tohoto řešení je, že není možné zamezit přístup pouze určitému uživateli, který ohrožuje Skrytou službu, ale pouze všem uživatelům. To představuje velmi velké riziko pro bezpečný chod a dostupnost serveru.

Složení adresy je tvořeno základní .onion adresou serveru a 22 znaků pro dešifrování IP.

```
wnv3zrtcklmlkprx.onion Cr+TM5Ii4QRHL5UMEE8TCQ
```

*Neviditelná „stealth“ autorizace*

Druhým protokolem pro ochranu je takzvaná neviditelná autorizace, která má stejné vlastnosti jako předcházející základní, ale obsahuje zvýšenou ochranu. Ta eliminuje problém předcházejícího protokolu, kdy v případě vyzrazení symetrického klíče mohl kdokoliv přistupovat na server. Místo toho se aplikuje uživatelský přístup, kdy Skrytá služba může vygenerovat až 16 jedinečných asymetrických .onion adres a symetrických descriptor cookies. Díky tomu má každý uživatel svoji vlastní adresu pro přístup na server.

V případě, že by jedna z adres byla vyzrazena, můžeme na serveru jednoduše tento klíč odebrat, vytvořit jiný a to bez ovlivnění zbývajících uživatelů.

Díky nutnosti vytvářet pro každého uživatelé soubor descriptor-id a .onion adresu, které jsou následně rozeslány do databází, je nutné omezit počet těchto adres. Maximum adres, jak již bylo zmíněno je 16 což má za následek nevhodnost řešení pro velký počet přistupujících uživatelů.

Druhou možností je distribuovat tyto klíče skupinám lidí místo jednotlivým uživatelům. Můžeme tak bez nutnosti další autentifikace limitovat nebo naopak rozšiřovat přístup skupině uživatelů.

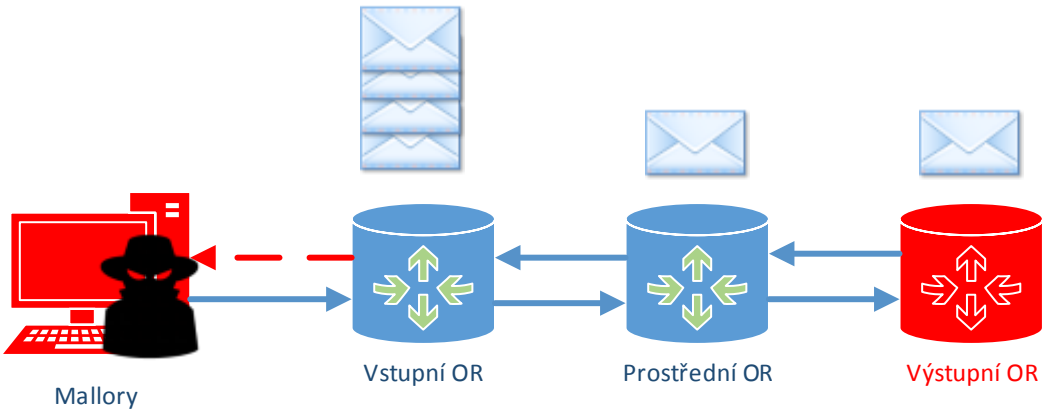
## Útok odstřelovače

Počátkem roku 2014 byl uveřejněn účinný útok zvaný „The Sniper Attack“ [11, 12] neboli útok odstřelovače, který by byl, v době testování, schopen poškodit dvacet nejpoužívanějších výstupních Tor směrovačů a to v rozmezí několika hodin útokem na vyrovnávací paměť jednoho z cibulových směrovačů. Zároveň tento útok mohl být použit ke zjištění totožnosti libovolné skryté služby.

Cílem tohoto útoku je docílit přetečení vyrovnávací paměti jednoho směrovače zneužitím chyby v řízení front, který používá síť Tor.

Komunikace je složena z klienta, který vytvoří okruh s prvním vstupním cibulovým směrovačem zvaným Tor Guard (ochránce), ten následně sestaví okruh s prostředním směrovačem a následně s posledním odchozím směrovačem. Při takto sestaveném spojení poslední odchozí směrovač vytvoří počítadlo s výchozí hodnotou 1000 buněk. Při příchodu dat tento směrovač data zabalí, sníží hodnotu čítače o jednu buňku a odešle ji skrze zbylé směrovače ke klientovi. Klient naopak za každou přijatou buňku inkrementuje svůj čítač o hodnotu jedna. Jakmile tato hodnota dosáhne 100 přijatých buněk, OP odešle zprávu SENDME, která oznámí koncovému směrovači, aby inkrementoval hodnotu svého čítače o 100.

Útok odstřelovače pracuje na principu DoS útoku, při kterém útočník využívá vlastnosti Toru, který funguje na protokolu TCP, zaručující doručení zpráv v pořadí a to odmítnutím číst příchozí zprávy od vstupního směrovače a zasíláním zpráv SENDME výstupnímu směrovači s požadavkem na další zprávy. To má za následek, že zprávy jsou odeslány na vstupní směrovač, který začne ukládat zprávy do té doby, než dojde k přetečení vyrovnávací paměti a následném ukončení procesu operačním systémem, na němž běží směrovač.




**Obr. 2.13. Útok odstřelovače**

Existují dvě varianty tohoto útoku. První využívá útoku pomocí klienta a ovládaného výstupního směrovače ke „zničení“ vstupního směrovače. V druhé typu útoku není nutné vlastnit jeden z Tor směrovačů, stačí pouze vybudovat spojení s cílovým serverem a zaslat na něj data. Ten tato data odmítne přijmout a klient pošle další zprávu SENDME a data, která zahltí výstupní směrovač.

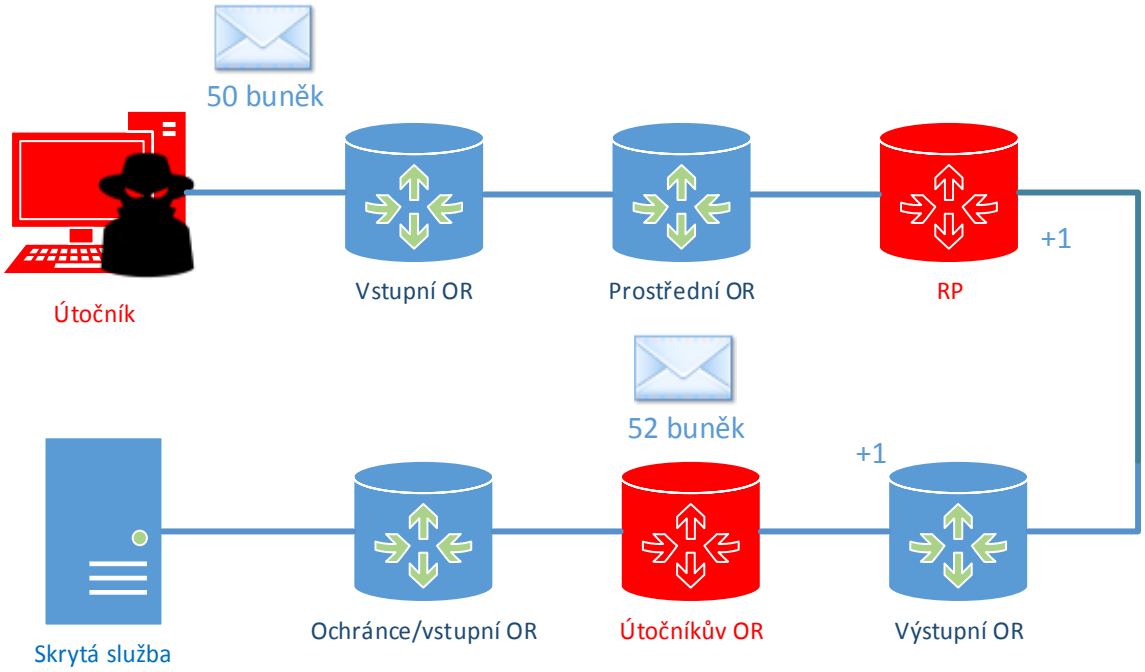
Síť Tor má způsoby, jak se bránit proti zneužití fronty a vícero požadavkům SENDME. Pokud útoční překročí hodnotu okna nastavenou na maximální hodnotu 1000, směrovač pošle zprávu DESTROY, která má za cíl ukončit spojení na všech sestavených okruzích s klientem. Prostřední směrovač zprávu přijme a přepošle ji vstupnímu. Vstupní směrovač má za cíl tuto zprávu přeposlat klientovi, který při obdržení zprávy zašle zprávu zpět o zrušení okruhu. Díky tomu se uvolní spojení a vstupní směrovač může vyčistit svoji frontu. V tomto případě však klient nečte příchozí zprávy a spojení mezi klientem a vstupním směrovačem není ukončeno a data jsou stále ve vyrovnávací paměti směrovače, zároveň útočník může kontrolovat velikost okna a vyhnout se detekci zasíláním příliš velkého množství SENDME zpráv s krátkém časovém úseku.

## Útok na ochránce Skryté služby

Tento útok může být kromě zamezení dostupnosti zneužit i k prolomení anonymity a to kontrolou vstupního a výstupního bodu komunikace.

Skrytá služba si v základním nastavení volí tři ochránce, kteří slouží jako vstupní body komunikace. Tyto směrovače mají zvýšené požadavky na to, aby dosáhly tohoto označení. Je nutná dostatečně dlouhá doba aktivity, dostatečná kapacita linky a jiné. Ochránci se, v rámci bezpečnosti, každý měsíc až dva mění, aby bylo dosaženo lepší bezpečnosti. Cílem útoku je zjistit ochránce, které používá Skrytá služba, zničit je a nahradit je vlastním ochráncem.

V první kroku je nutné, aby útočník vlastnil jakýkoliv směrovač, který bude sloužit jako bod seznámení (RP) a jeden směrovač jako ochránce. Útočník, v podobě klienta, donutí sestavit Skrytou službu nové okruhové spojení skrze požadavek RP a to tolikrát až je zvolen útočníkův ochránce jako jeden ze skoku. Následně RP odešle zprávu PADDING o velikosti 50 buněk Skryté službě. Útočníkův ochránce následně detekuje vzorec, ze kterého jsou patrné 2 zprávy od RP a celkově 52 zpráv směrem ke Skryté službě, které jsou následované zprávou DESTROY. Díky tomu útočníkův směrovač zjistí, že je dva skoky před Skrytou službou a jeden skok před jeho ochráncem. Pomocí několika nových požadavků je schopen tento směrovač detekovat všechny tři ochránce Skryté služby.




**Obr. 2.14. Útok na Ochránce Skryté služby**

Druhým krokem je zaútočit na tyto směrovače pomocí paralelních útoků Odstřelovače a to na všechny objevené směrovače s cílem vyřadit je z provozu.
Posledním krokem je nechat Skrytou službu zvolit si nového ochrance. V případě, že útočníkův ochránce bude zvolen, zvýší se počet buněk od RP na 3 a celkový počet bude 53 buněk před zprávou na zrušení okruhu. Díky tomu ochránce zná přesnou adresu Skryté služby. V případě, že směrovač nebude zvolen jako ochránce, útočník bude pokračovat v útocích opět od prvního kroku do té doby, než uspěje.

*Ochrana před útokem Odstřelovače*

Jednou z možných ochran před tímto útokem je vyžadování odpovědi od klienta po každé sté buňce s vypočítáním haše a odesláním odpovědi v SENDME buňce. V případě špatné odpovědi bude spojení ukončeno. Tento způsob však neřeší paralelizaci tohoto útoku, který by byl pouze ztížen, ale ne zamezen.

Dalším možností je limitování velikosti, která je už částečně implementována. Pokud fronta překročí hodnotu tisíce buněk, bude automaticky vyprázdněna a okruh přerušen.

Obě řešení je vhodné doplnit o zrušení nejstarší fronty v případě, že dojde k přetékaní vyrovnávací paměti. Díky tomu bude vyprázdněna fronta, která nejdelší dobu v nečinnosti a uvolní tak prostředky směrovače.

Všechna tato řešení již byla zaslána správcům sítě Tor, kteří je implementovali do nové verze 0.2.4.18-rc. Díky tomu by měla být eliminována tato velmi závažná bezpečností chyba a velmi ztížena možnost zjištění identity Skryté služby.

## 2.4. Anonymní autentizace vůči serveru

V předcházejících částech práce byly popsány a řešeny způsoby jakými anonymně přistupovat na vzdálený server. Následující část se zaměří na anonymní autentizaci vůči serveru. Spojením těchto dvou částí bude dosaženo komplexního návrhu řešení anonymního pohybu po Internetu.

Název této kapitoly obsahuje slova anonymita a autentizace, která tvoří dva protiklady a vzájemně se téměř vylučují. Není totiž možné chtít se pohybovat po internetu či serveru zcela anonymně a zároveň požadovat přístup na privátní části serverů. Zjednodušeně řečeno není možné, aby uživatel elektronické banky, jen tak přistoupil do svého bankovnictví a byl bez jakýchkoliv identifikačních znaků puštěn právě do svého a ne k účtu někoho jiného nebo naopak.

Na druhou stranu vyžadování autentizace, tedy ověřováním identity, ztrácí člověk svoji anonymitu, protože ztrácí pocit bezpečí. Toto je běžný jev ztráty soukromí, ze kterého má člověk pocit jistého vpádu do soukromí a ztráty kontroly. Tento nepříjemný pocit může vést k pocitu, že někdo o nás ví mnoho informací. V dnešní době sociálních sítí tedy není problém zjistit o každém mnoho informací a to často nevědomky, zveřejňováním a vyplňováním běžných informací v podobě kde žijeme, kde jsme se narodili atd. Dalším zdárným příkladem, může být cílená internetová reklama. Ta identifikuje náš pohyb po mnoha internetových stránkách, snaží se analyzovat, co nás zajímá, předpovědět co budeme následně dělat a to vše za účelem cílené reklamy, které zvýší prodeje nabízených výrobků. Vše se ve většině případu děje nepozorovaně, což může být velmi lehce zneužitelné.

Jak bylo popsáno ve výše zmíněných odstavcích, nelze jednoduše navrhnout komplexní systém plně anonymního přístupu ve spojení s ověřením totožnosti uživatele. Je nutné vytvořit kompromis proti oběma pólům tak, aby uživatelé měli stále pocit anonymity a zároveň mohla být ověřena jejich identita.

### 2.4.1. Zabezpečení přenosu prostřednictvím protokolu HTTPS

Jednou z prvních a velmi důležitých částí pro návrh je bezpečnost, která vytváří důvěru mezi komunikujícími stranami. Ta je dále nutná k tomu, aby se předešlo možným podvrhům, sledováním a cíleným útokům na datovou komunikaci.

Představitelem tohoto zabezpečení, především pro internetovou komunikaci mezi uživateli a servery, je protokol HTTPS. HTTPS je zkratka z anglického Hypertext Transfer Protocol Secure. Jedná se v podstatě o klasický protokol HTTP, běžícím na portu 80, přes který běžně navštěvujeme různé webové stránky, avšak poskytuje vrstvu

zabezpečení, která je vhodná pro výměnu citlivých dat. HTTPS využívá pro šifrování dat protokoly SSL nebo TLS a mění obvykle port komunikace na standardní 443.

Použití protokolu HTTPS má dvě hlavní využití. Tím prvním je šifrování, které má na starost skrytí přenášených dat, změnou posloupnosti znaků, což ztíží nebo znemožní přečtení originální zprávy. Druhým účelem je identifikace, pomocí které jsme schopni ověřit, s kým komunikujeme a jestli dané osobě, počítači či serveru můžeme důvěřovat, že je tím za koho se vydává.

Existují dva způsoby šifrování, jímž jsou symetrické a asymetrické šifry. První zmíněná šifra spoléhá na to, že obě strany komunikace znají stejné tajemství, kterým zašifrují a následně dešifrují přenášená data.

Asymetrická šifra pracuje na principu dvou klíčů, kde jedna strana poskytne druhé svůj veřejný klíč, kterým zašifruje komunikaci, ale zároveň si ponechá privátní klíč, pomocí něhož je schopna zprávu dešifrovat. Pro pochopení uveďme příklad.

Předpokládejme, že jsme uživatelé emailové schránky, kteří se chtějí připojit na poštovní server a přečíst si poštu. Aby naše komunikace byla bezpečná, použijeme symetrický klíč, poskytnutý serverem, pomocí kterého zabezpečíme zprávu. Server si tuto zprávu pomocí téhož klíče dešifruje. Takto zabezpečená zpráva je sice zabezpečeně přenesena, ale je dosti nepraktická a nebezpečná. V případě vice než jednoho uživatele by nebyl žádný problém použít jeden klíč k dešifrování zpráv jiných uživatelů. Nemluvě o obtížném a bezpečném přenosu tohoto klíče. Z těchto důvodů se používá asymetrické šifrování. Server poskytne nám a všem ostatním uživatelům veřejný klíč, jenž je pro všechny stejný. Pomocí něj zprávu zašifrujeme a zašleme serveru. Server je vlastníkem jediného privátního klíče, kterým je schopen zprávu dešifrovat.

### 2.4.2.Certifikační autorita

Protokol HTTPS slouží krom šifrování i k identifikaci komunikujících stran. K tomu je využito tzv. Certifikačních Autorit (CA), které podepisují veřejné šifrovací klíče v podobě certifikátů. Podepsáním certifikátu CA zaručuje autenticitu a správnost údajů obsažených v certifikátu.

CA je úměrně závislá a bezpečná podle správnosti uvedených dat. V případě, že by CA podepsala certifikát, který by byl falešný, ztratila by na své důvěryhodnosti. To by mělo za následek ztrátu důvěry ke všem certifikátům podepsaných touto autoritou. Z těchto důvodů si nechává CA velmi záležet na správnosti poskytnutých údajů.

Postup ověření je následující:

1. Společnost nebo jednotlivec požádá certifikační autoritu o certifikát. Existují různé společnosti po celém světě příkladem VeriSign, RapidSSL, GeoTrust a jiné. Těmto CA je předán název webového serveru, o jakou společnost se jedná, kde se nachází atd.

2. CA ověří tyto informace v různých veřejných rejstřících a zašle dopis na udanou adresu.
   Z těchto dat a vygenerované žádosti z privátního klíče CA vytvoří certifikát a podepíše jej. Díky tomu je zaručena nemožnost pozměnit uvedená data. Certifikát obsahuje verzi, sériové číslo, ID algoritmu, platnost, informace o společnosti, veřejný a osobní klíč, identifikátory a podpisový algoritmus z haše všech těchto informací.
3. Společnost obdrží vytvořený certifikát, který může implementovat na svůj server.
4. Webový prohlížeč při přístupu na server ověří certifikát u jedné z těchto společností.
5. Prohlížeč důvěřuje certifikátů a umožní komunikaci.

V případě neveřejných částí internetu nebo spíše intranetu je možné vytvořit svoji vlastní certifikační autoritu. Tato autorita bude mít stejné vlastnosti jako veřejné CA, problém však spočívá v důvěře. Záleží pouze na nás, zdali budeme věřit, že tato CA podepisuje korektní certifikáty. Přidáváním těchto CA zvyšujeme riziko možné diskreditace a tím riziko důvěryhodnosti a možných hrozeb.

### 2.4.3. TLS/SSL

Jak bylo popsáno, HTTP protokol využívá dva kryptografické protokoly, jimiž jsou SSL a TLS.

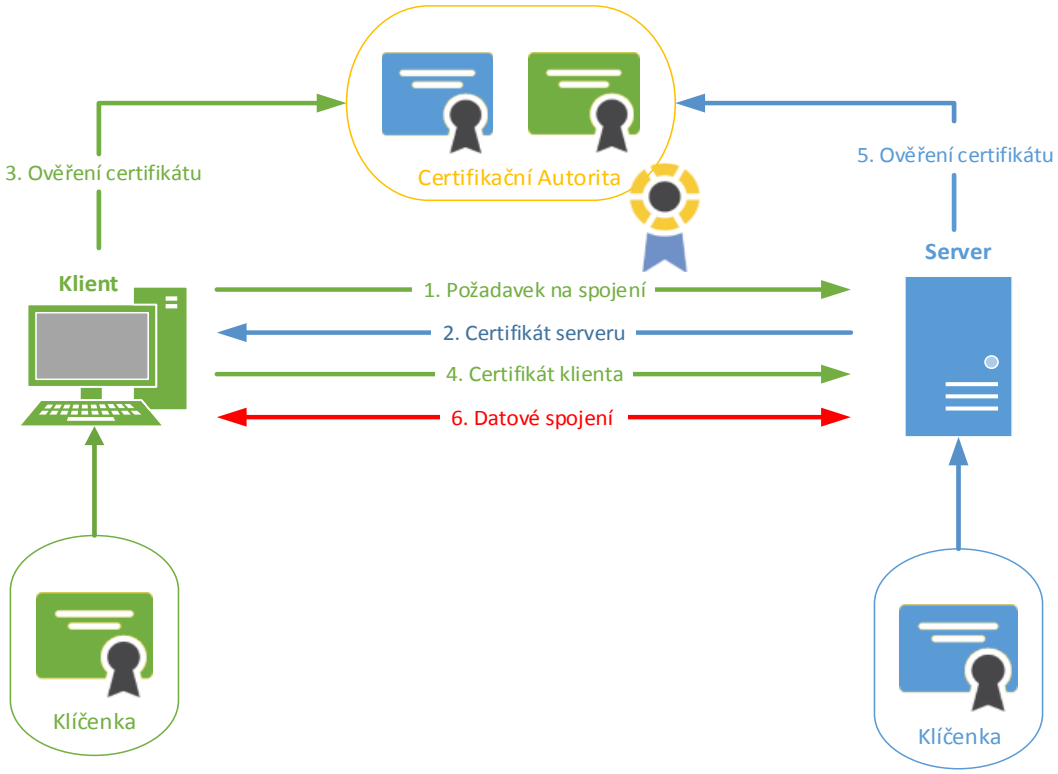
Prvním z nich je SSL, neboli Secure Socket Layer, tedy zabezpečená vrstva mezi transportním protokolem TCP a aplikační vrstvou v podobě například protokolu HTTP. Tento protokol byl vyvinut společností Netscape v roce 1994, z nutnosti zabezpečit přenos dat v nově vzniklém grafickém webovém prohlížeči. Tento protokol se stal neoficiálním standardem z důvodu jeho velmi rozšířeného použití.

V roce 1999 byl vydán nový otevřený standard TLS (Transport Layer Security), který sjednocuje kryptografické protokoly. Jeho základem se stal protokol SSL ve verzi tři. Z tohoto důvodu se často uvádí spojení SSL/TLS, jelikož oba protokoly jsou téměř totožné a TLS je zpětně kompatibilní s předchozím SSL. TLS obsahuje silnější šifrovací algoritmus a může pracovat i na jiném portu než výchozí 443. Oba protokoly podporují jak asymetrické, tak symetrické klíče.

Pro navázání zabezpečeného spojení slouží takzvaný „handshake“ v češtině potřesení rukou. Tento proces potřesení rukou slouží k ustanovení důvěryhodného spojení mezi dvěma entitami za využití jednosměrné nebo obousměrné autentizace.

Z abstraktního pohledu při navazování spojení je možné rozdělit sestavování následovně [8]:

1. Klient zašle požadavek na sestavení zabezpečeného spojení.
2. Server zašle svůj certifikát pro ověření identity.
3. Klient ověří certifikát u důvěryhodné certifikační autority.
4. V případě obousměrné autentizace předloží klient svůj certifikát.
5. Server taktéž ověří certifikát u CA.
6. Klient se serverem vytvoří zabezpečené a důvěryhodné spojení.



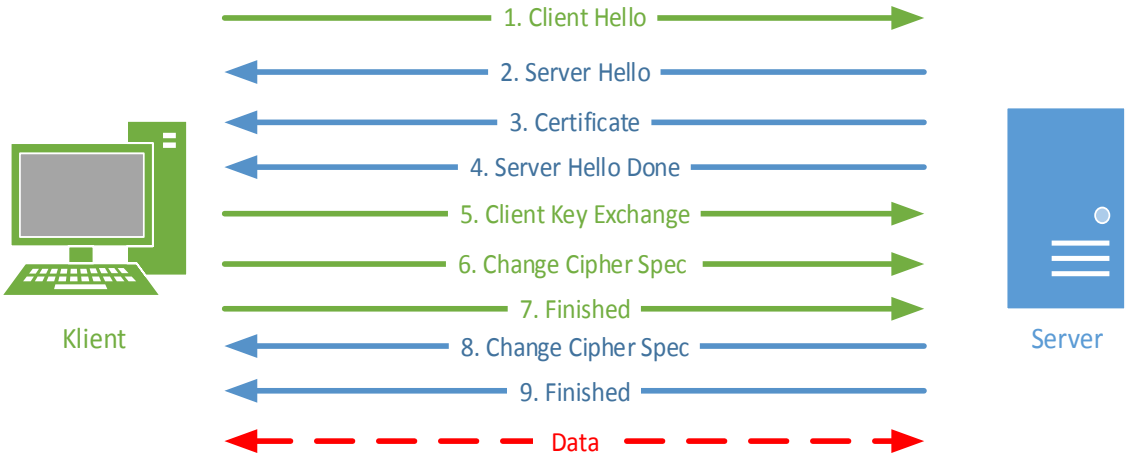

**Obr. 2.15. Navázaní TLS/SSL spojení**

## Jednosměrná SSL autentifikace

Jednosměrná autentifikace je nejpoužívanější metodou pro navázání důvěryhodného spojení mezi klientem a serverem. Server nezajímá, kdo na něj přistupuje, ale naopak klient ověřuje, zdali je server opravdu tím, za koho se vydává a zdali mu lze důvěřovat.

Celý proces jednosměrné SSL autentizace je rozdělen do devíti zpráv nutných k sestavení spojení [9].

1. Klient zašle „Client Hello“ zprávu společně se všemi podporovanými kryptografickým algoritmy, jako verzi, podporované šifry a hash metodu. Zpráva taky obsahuje 32 bytové číslo, kde 4 byty slouží k přenosu času a data klienta a dalších 28 bytů je použito pro náhodně vygenerované číslo.
2. Server odpoví zprávou „Server Hello, která obsahuje stejně jako předchozí zpráva 32 bytové číslo, které bude následně s klientským náhodným číslem použito k vygenerování hlavního tajného klíče (Master Secret). Dále je ve zprávě obsažena nejbezpečnější ze všech společných šifer, tak aby bylo dosaženo co nejlepšího zabezpečení. V případě, že obě strany nebudou podporovat stejné šifry, bude spojení ukončeno.
3. Následně server zašle další zprávu „Certificate“, která obsahuje certifikát serveru, ve kterém se nachází veřejný klíč, který bude použit pro šifrování spojení. Server odpoví „Hello done“ zprávou a čeká na odpověď.
4. Poslední zprávou před odpovědí klienta je „Server Hello Done“, která oznamuje ukončení vysílání zpráv ze strany serveru a čekání na zprávy klienta.
5. Zatím co server zasílá tuto zprávu, klient kontroluje platnost certifikátu u CA a platnost „Server Hello“ zprávy serveru. Klient následně zašle zprávu „Client Key Exchange“, která obsahuje předběžný hlavní klíč, který je zašifrován veřejným klíčem serveru. Tento klíč je vypočítán díky oběma náhodným číslům serveru a klienta. Klient se zároveň ujistí, že server je tím, za koho se vydává, jelikož pouze on má privátní klíč k dešifrování zprávy. Další součástí této zprávy je verze protokolu, která ověří že se používá šifra, která byla již domluvena. Toto opatření předchází útoků, které by mohly zmanipulovat některou ze zpráv a snížit tak šifru na některou z méně bezpečných.
6. Klient zašle zprávu „Change Cipher Spec“ ve které se říká, že po odeslání zprávy „Finished“ budou použity domluvené nové klíče a šifrování navázaného spojení.
7. Zprava „Finished“ říká serveru, že nastavení ze strany klienta je hotové a klient je připraven na šifrované spojení.
8. Server ověří zaslaná nastavení, klíče a odpoví zprávou potvrzující změnu šifrování „ChangeCipherSpec“
9. Server odešle poslední zprávu „Finished“ před zahájením šifrovaného spojení.




**Obr. 2.16. Jednosměrná SSL autentifikace**

## Obousměrná SSL autentifikace

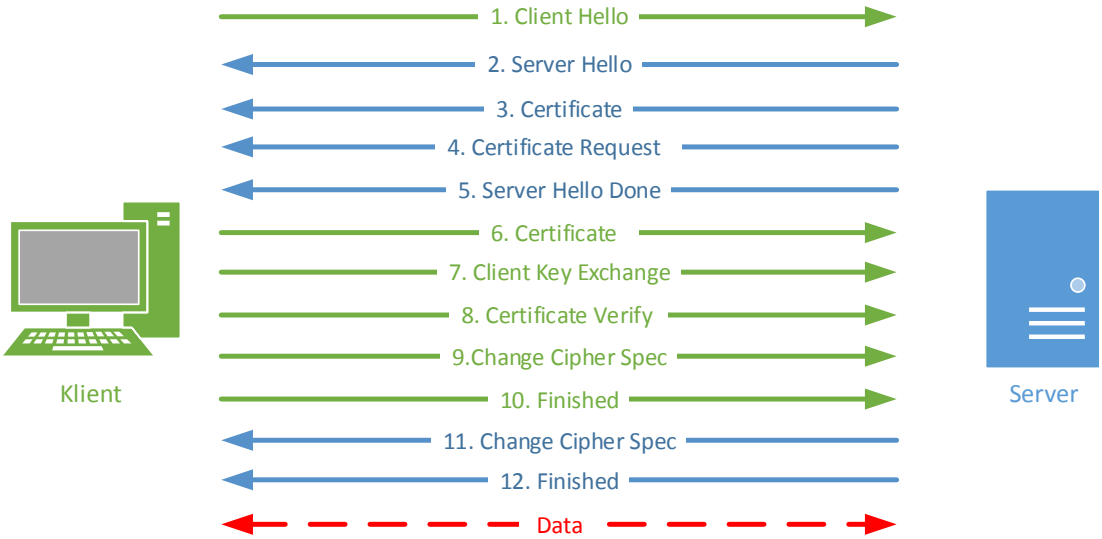
Obousměrná SSL autentifikace je oproti jednosměrné používaná minoritně a to především z důvodu nutnosti klienta vlastnit osobní certifikát a dále z důvodu výpočetní náročnosti při sestavování spojení. Použití obousměrné autentifikace je vhodné tam, kde oběma stranám komunikace záleží vzájemném ověření totožnosti.

Další podstatnou výhodou je možnost ověření klienta vůči serveru a to bez nutnosti jakkoliv přenášet heslo k ověření vůči serveru. Tím odpadají obavy o možné odposlechnutí či podvržení hesla. Nevýhodou tohoto řešení je způsob zabezpečení na straně klienta. Při normálním využití hesla spoléháme na znalost tajné informace, při čemž v případě klientského certifikátu spoléháme na jeho vlastnictví.

Způsob potřesení rukou pro sestavení spojení je téměř totožný s tím jednosměrným. Celkový počet zpráv se zvyšuje na dvanáct.

1. Klient zašle zprávu „ClientHello“ jako požadavek na sestavení spojení.
2. Server odpoví zprávou „ServerHello“ s výběrem nejlepší šifry a náhodným číslem.
3. Server odešle zprávu „Certificate“ obsahující veřejný klíč serveru a další informace.
4. Následně server odešle zprávu „ClientCertificeteRequest“ ve kterém server požaduje, aby se klient autentizoval pomocí svého certifikátu.
5. Poslední zprávou je „ServerHelloDone“.
6. Klient odpoví zprávou „Certificate“ obsahující klientův certifikát.

7. Následuje zpráva „ClientKeyExchange“, která je zašifrována veřejným klíčem serveru.
8. Další zpráva zaslaná klientem je „CertificateVerify“, ve které se klient autentizuje zašifrováním haše všech zaslaných zpráv svým privátním klíčem. Server po obdržení této zprávy použije pro dešifrování klientův veřejný klíč, obsažený v poskytnutém certifikátu a tím potvrdí svoji totožnost.
9. Klient zašle zprávu „ChangeCipherSpec“.
10. Zpráva „Finished“ říká serveru, že nastavení ze strany klienta je hotové a klient je připraven na šifrované spojení.
11. Server ověří zaslaná nastavení, klíče a odpoví zprávou potvrzující změnu šifrování „ChangeCipherSpec“.
12. Poslední zprávou před začátkem šifrovaného spojení je zpráva „Finished“.




**Obr. 2.17. Obousměrná SSL autentifikace**

## Útok heartbleed

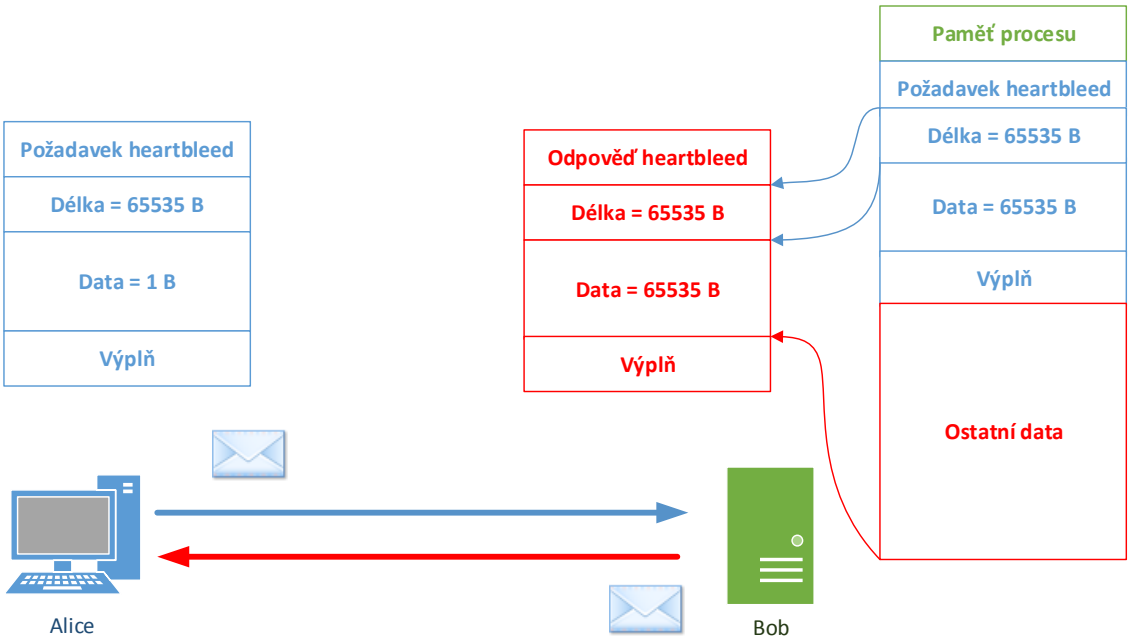
Chyba heartbleed [22] neboli krvácející srdce je jednou z největších bezpečnostních chyb v historii Internetu, která využívá chyby v knihovnách OpenSSL a která umožňuje číst data v paměti serveru.

Tato chyba byla představena koncem roku 2011 a implementována 14. března 2012 do knihovny OpenSSL v. 1.0.1. Její pojmenování vychází z implementace heartbeat (tlukoucí srdce) rozšíření, které bylo přidáno do TLS/DTLS spojení.

Význam heartbeat rozšíření je z důvodu udržování zabezpečeného spojení mezi klientem a serverem a to i přes nespolehlivé protokoly jako UDP. Heartbeat [4]

v podstatě funguje jako ICMP zpráva, která se dotazuje, zdali je druhá strana dostupná a tím udržuje navázané spojení i v případech, kdy se zrovna nepřenáší žádná data.

Chyba v knihovně SSL se projevuje při zaslání dotazu serveru. Tato zpráva obsahuje v hlavičce délku zprávy, tvořenou dvěma bajty a následovanou přenášenými daty a případnou výplní. Server, který tuto zprávu příjme, odpoví stejnou zprávou a odešle stejné množství dat, jaké byly přijaty, tedy zmíněnou délku zprávy. Chyba však spočívá v neověřování přijatých dat, kde se neporovnává délka skutečně přijatých dat. Toho může být zneužito vytvořením zprávy o deklarované délce 65535 bajtů a pouze jedním bajtem skutečných dat. Server, který takovouto zprávu přijme, vytvoří odpověď, která bude o délce 65535 bajtů, bude obsahovat jeden bajt přijatých dat a zbytek zprávy bude obsahovat obsah paměti, která přímo následuje.




**Obr. 2.18. Útok heartbleed**

Tato část paměti může obsahovat v podstatě libovolná data. V základu je lze rozdělit na tři druhy. Tím prvním a nejkritičtějším jsou primární klíče, které slouží k šifrování celé komunikace a které mohou být zneužity k útoku MitM. Druhý typ dat může obsahovat uživatelská jména a přístupová hesla ke službám. Poslední typ dat může obsahovat osobní data, jako emailové zprávy, důležité dokumenty a mnohé jiné.

Další zákeřností kromě ztráty citlivých dat je, že není možné ověřit, jestli k úniku či pokusům o prolomení došlo.

Ochrana spočívá v implementaci nových knihoven OpenSSL v 1.0.1f vydaných 7. dubna 2014 opravující heartbleed chybu a následné revokaci vydaných certifikátu a jejich opětovném vydání. To vše doprovázené i o změnu přístupových hesel pro uživatele, která mohla být odposlechnuta. Chyba heartbleed zároveň nemusela ovlivnit TCP spojení a protokoly jako HTTPS, jelikož ty nutně tyto zprávy nepoužívají. Dále je také možné tyto dotazy blokovat a zakázat heartbleed rozšíření.

# 3. Implementace anonymního přístupu

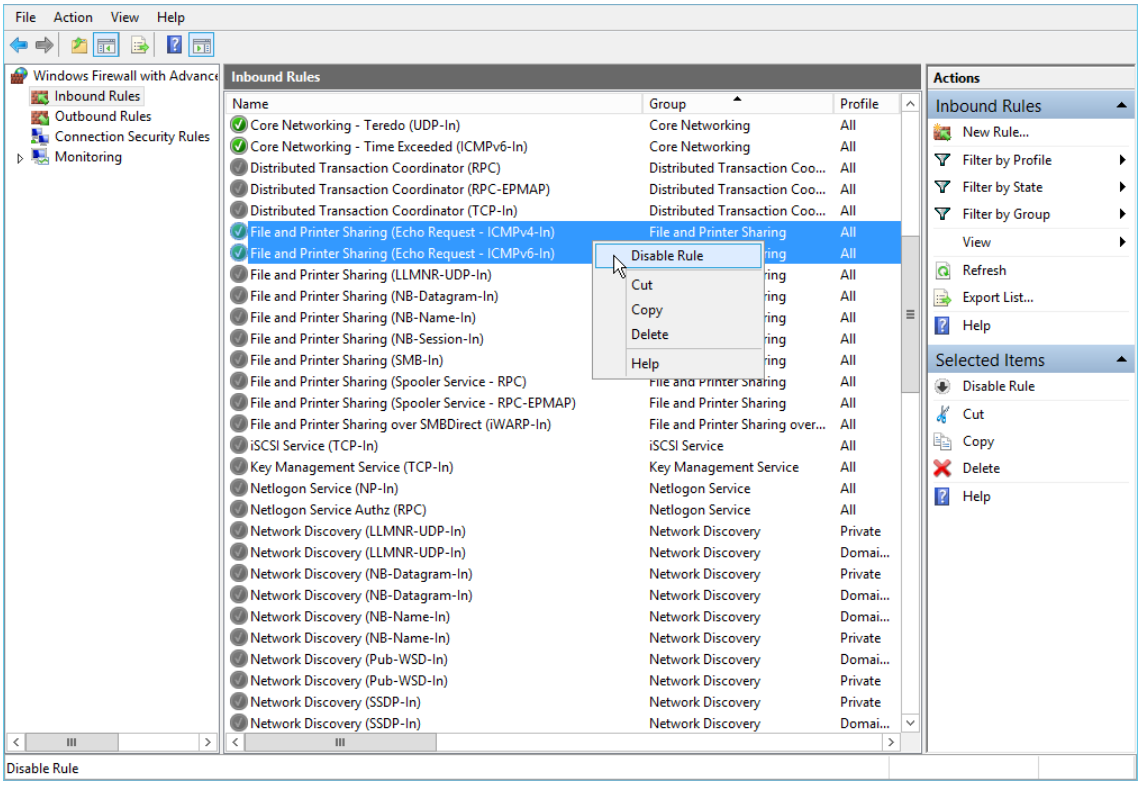
## 3.1. Nastavení ICMP zpráv

Jedním z již zmíněných útoků vůči klientovi je ověřování dostupnosti klienta a jeho klientské aplikace. K zamezení této kontroly je nutné aplikovat pravidlo Firewallu na cílové stanici tak, aby neodpovídala na dotazy „ICMP echo request“.

Toto pravidlo je defaultně nastaveno na všech serverech společnosti Microsoft, avšak u distribucí pro osobní počítače Windows XP, 7, 8, 8.1 a jiné, je toto pravidlo ve výchozím nastavení povoleno.

Pro nastavení brány firewall v systému Windows je tedy nutné postupovat následovně:

1. Spustit konzoli klávesovou zkratkou „Win+R“.
2. Zadat příkaz „wf.msc“ a potvrdit.
3. Následně se otevře okno nastavení firewallu systému Windows.




**Obr. 3.1. Nastavení firewallu systému Windows**

4. Přesuneme se do nastavení příchozích pravidel (Inbound Rules) a nalezneme pravidlo pro nastavení ICMP zpráv „File and Printer Sharing (Echo Request – ICMPv4-In)“ a „File and Printer Sharing (Echo Request – ICMPv6-In)“. Obě tato pravidla nastavují, jak bude zacházeno se zprávami „ICMP echo request“ a to pro oba síťové protokoly IPv4 a IPv6.
5. Následně je nutné pravidla zakázat v případě, že byla povolena.

## 3.2. Instalace Toru

### 3.2.1. Druhy balíčků

Projekt Tor poskytuje ke stažení mnoho balíčků s prvky nutnými pro běh Toru na uživatelském zařízení. Jelikož je program multiplatformní, je možné stažení balíčků pro Microsoft Windows, Apple OS X, Linux/Unix, tak i pro chytré telefony na platformě Android.

*Tor Browser Bundle:*

Tento balíček je dostupný pro všechny operační systémy a poskytuje kompletní balík nástrojů pro anonymní pohyb po Internetu. Je složen ze samostatných zdrojových kódů Toru, grafické nadstavby Vidalia, která poskytuje velké množství informací a umožňuje nastavit chování OP. Další důležitou součástí je upravený prohlížeč Firefox pro anonymní pohyb. Ten je přednastaven tak, že zakazuje použití Flashe a spouštění skriptů. Zároveň je rozšířen o doplněk HTTPS Everywhere, který vynucuje použití zabezpečeného protokolu HTTPS při přístupu na webové stránky.

*Vidalia Bridge Bundle:*

Balíček Vidalia Bridge slouží jako mostové spojení ostatních uživatelů skrze Tor k Internetu. ISP může zablokovat volně dostupný seznam výstupních směrovačů a tím uživateli zamezit přístup do Toru. Díky této verzi může uživatel přidat přístupové body, které nejsou součástí rejstříku přístupových bodů a nemusí být neustále dostupné. Pro získání adres těchto přemosťujících spojení je nutné nastavit v aplikaci potvrzení blokace Internetu, navštívit stránky [3] nebo napsat email na bridges@bridges.torproject.org a do zprávy umístit text „get bridges“. Aby útočník nebyl schopen získat kompletní seznam mostových spojení, je nutné v případě webové stránky použití CAPTCHA kódů a v případě emailů pouze domény gmail.com.

*Vidalia Relay Bundle:*

Balíček, pomocí něhož může uživatel pomoci rozšířit síť Tor o další cibulový směrovač, který je použit pouze jako prostředník mezi dalšími OR a nekompromituje tak veřejnou adresu uživatele.

*Vidalia Exit Bundle:*

Totožný balík jako Relay Bundle, avšak umožňuje sdílet své Internetové spojení jako bránu do Internetu a to se všemi právními důsledky v případě zneužití.

*Expert Bundle:*

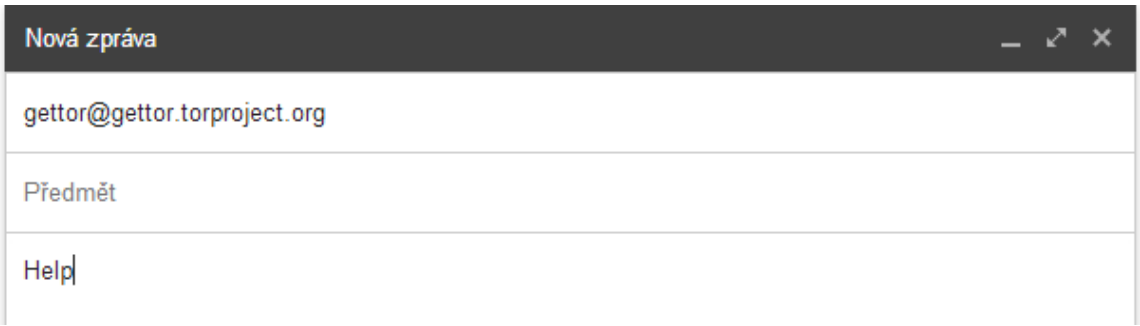
Balík s čistým Torem, který je nutné nakonfigurovat pro jednotlivé služby.

### 3.2.2. Získání balíčku Tor

Pro prvotní získání balíčku s aplikací Tor je možné navštívit webové stránky projektu na adrese [24], ze kterého je možné stáhnout instalační balíček Toru. Na tyto stránky je doporučeno přistupovat přes zabezpečený protokol HTTPS, který zaručuje bezpečnou komunikaci a společně s vydaným certifikátem, který je ověřen Certifikační autoritou potvrzující identitu cílové webové stránky Toru. Bohužel ani to nezaručuje úplnou důvěryhodnost této stránky, která mohla být v minulosti ohrožena [10].

V případě, že webové stránky projektu Tor jsou blokovány vládou nebo ISP, je možné využít síť dobrovolníků, kteří instalační balíčky zrcadlí na své weby [27].

Další možností jak získat program Toru je využití jednoho z velkých poskytovatelů emailových schránek, který umožňuje přijímat velké soubory. Příkladem budiž emailová schránka od společnosti Google. V případě odeslání emailové zprávy na gettor@gettor.torproject.org a zadáním slova „Help“ do těla zprávy, obdržíme po odeslání zprávu, obsahující návod a seznam dostupných balíčků Tor.



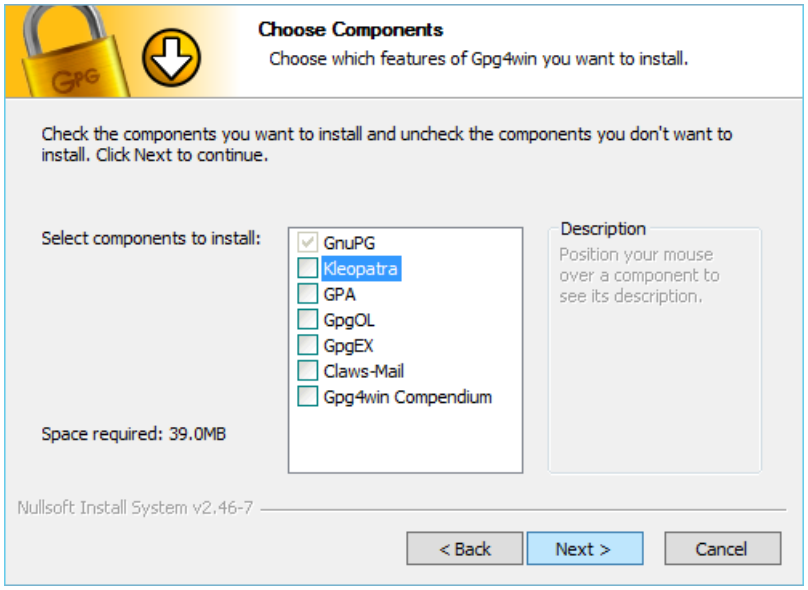

**Obr. 3.2. Tělo zprávy pro získání Toru skrze email**

### 3.2.3. Ověření obdrženého balíčku

Dalším z již zmíněných možných útoků je falešný program Tor, který by mohl být upraven a následně poskytnut pro další použití. Takto upravený klient by sice mohl být připojen do sítě Tor, ale zároveň by mohl odesílat citlivé informace o samotném uživateli a tím ho zkompromitovat. U klasických programů šířených po Internetu se kontroluje, zdali odpovídá kontrolní součet. Tento součet informuje, jestli byl balíček stažen bez chyb, avšak neověřuje autenticitu tohoto programu.

Z tohoto důvodu jsou všechny balíčky dostupné se soubory mající stejný název jako stahovaný balíček a navíc mají příponu „.asc“. Tyto soubory slouží jako podpis otevřeného projektu GnuPG a tím pomáhají ověřit, kdo daný balíček podepsal.

Pro ověření je nutné stáhnout aplikaci Gpg4win pro systém Windows z [6] a nainstalovat ji. Není nutné instalovat ostatní rozšíření viz obrázek.




**Obr. 3.3. Instalace programu Gpg4win**

Po úspěšném nainstalování je dále nutné naimportovat veřejně dostupný klíč, který bude následně porovnáván s klíčem, který podepsal instalační balíček.

Pro naimportování klíče slouží následující příkaz:

```
"C:\Program Files (x86)\Gnu\GnuPg\gpg2.exe" --keyserver x-
hkp://pool.sks-keyservers.net --recv-keys 0x416F061063FEE659
```

Klíč je možné zobrazit pomocí příkazu:

```
"C:\Program Files (x86)\Gnu\GnuPg\gpg.exe" --fingerprint
0x416F061063FEE659
```

```
c:\Program Files (x86)\GNU\GnuPG>gpg2.exe --keyserver x-hkp://pool.sks-keyserver
s.net --recv-keys 0x416F061063FEE659
gpg: requesting key 63FEE659 from hkp server pool.sks-keyservers.net
gpg: key 63FEE659: "Erinn Clark <erinn@torproject.org>" not changed
gpg: Total number processed: 1
gpg:              unchanged: 1

c:\Program Files (x86)\GNU\GnuPG>gpg2.exe --fingerprint 0x416F061063FEE659
pub   2048R/63FEE659 2003-10-16
      Key fingerprint = 8738 A680 B84B 3031 A630  F2DB 416F 0610 63FE E659
uid                  Erinn Clark <erinn@torproject.org>
uid                  Erinn Clark <erinn@debian.org>
uid                  Erinn Clark <erinn@double-helix.org>
sub   2048R/EB399FD7 2003-10-16


c:\Program Files (x86)\GNU\GnuPG>
```

**Obr. 3.4. Naimportování klíče pro ověření podpisu balíčku Tor**

Nyní je již vše připraveno k tomu, abychom mohli porovnat oba podpisy a to za pomoci následujícího příkazu:

```
"C:\Program Files (x86)\Gnu\GnuPg\gpg2.exe" --verify c:\tor-
browser-2.3.25-13_en-US.exe.asc c:\tor-browser-2.3.25-13_en-
US.exe
```

```
c:\Program Files (x86)\GNU\GnuPG>gpg2.exe --verify c:\tor-browser-2.3.25-13_en-U
S.exe.asc c:\tor-browser-2.3.25-13_en-US.exe
gpg: Signature made 09/20/13 17:19:05 Central Europe Daylight Time using RSA key
 ID 63FEE659
gpg: Good signature from "Erinn Clark <erinn@torproject.org>"
gpg:                 aka "Erinn Clark <erinn@debian.org>"
gpg:                 aka "Erinn Clark <erinn@double-helix.org>"
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: 8738 A680 B84B 3031 A630  F2DB 416F 0610 63FE E659

c:\Program Files (x86)\GNU\GnuPG>
```

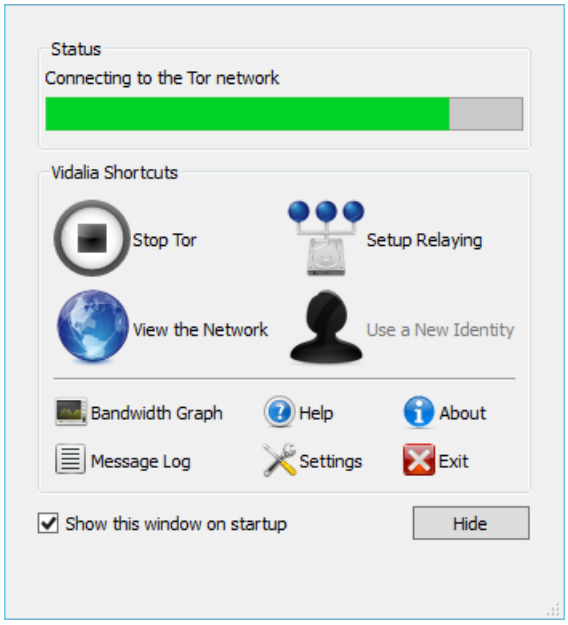
**Obr. 3.5. Porovnání klíče a digitálního podpisu aplikace**

Z výsledného porovnání je patrná shoda obou klíčů, avšak je zde uvedena důležitá informace o tom, že tento klíč není certifikovaný. Je tedy na našem uvážení, zdali budeme poskytnutému klíči důvěřovat.

### 3.2.4. Spuštění Toru

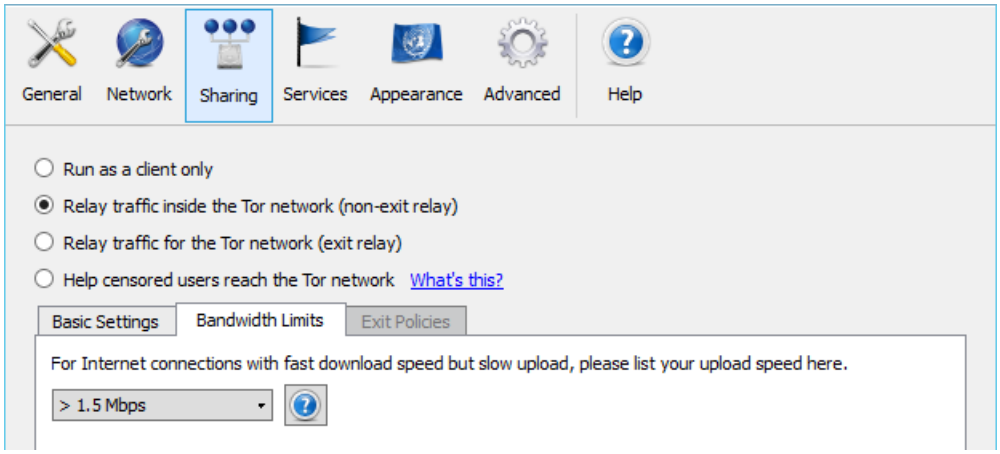
Po stažení přenosného balíku „Tor Browser Bundle“ a jeho extrahování, je možné spustit program zástupcem „Start Tor Browser“, který spustí ovládací prvek Vidalia a zároveň spustí předpřipravený prohlížeč Firefox. Po úspěšném připojení je zobrazena výchozí stránka s nově získanou IP adresu.

Jednou z dalších možností je instalace jednoho z balíků Vidalia, který se nainstaluje do složky *„C:\Program Files (x86)\“*, kde instalační složka je pojmenována dle stažené verze. Výhodou zmíněné instalační verze je také možný automatický start programu Vidalia při spuštění systému.




**Obr. 3.6. Hlavní okno programu Vidalia**

Z výše uvedeného obrázku je vidět ovládací panel Vidalia, který se snaží připojit do sítě Tor. Vidalia umožňuje zapnutí/vypnutí Toru, nastavení způsobu připojení do Tor sítě a dále pohled na mapu výstupních uzlů, grafické zobrazení datového provozu, tlačítko pro získání nové identity a tlačítko pro nastavení.




**Obr. 3.7. Nastavení programu Vidalia**

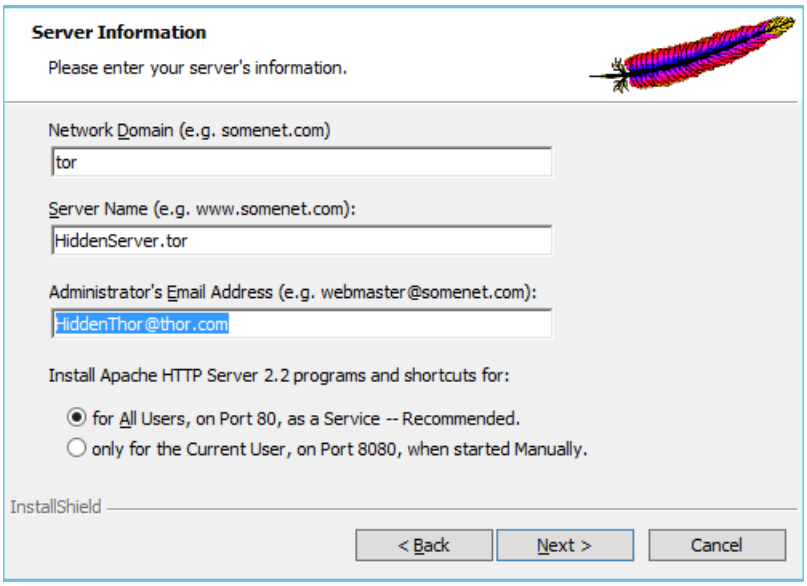
Nastavení programu Tor je koncipováno tak, aby bylo co nejjednodušší a nevyžadovalo odborné znalosti. Nejdůležitějším oknem nastavení je záložka sdílení, která umožňuje nastavit, jakým způsobem bude Tor použit. Tato nastavení korespondují s ostatními balíčky Vidalia, které jsou již přednastavené pro jednotlivé druhy spojení.

## 3.3. Instalace skryté služby

Pro demonstraci skryté služby byl zvolen Apache web server, který je jedním z nejrozšířenějších webových serverů a zároveň je multiplatformní, což koresponduje s verzemi samotného Toru.

Instalace Apache je poměrně intuitivní. Instalační soubory jsou dostupné ze stránek Apache Software Foundation [21] nebo Apache Lounge [1] pro systém Windows. Nejvhodnější verze je Apache doplněný o OpenSSL sloužící k zabezpečení spojení.

V průběhu instalace jsme vyzváni k zadání síťové domény, názvu serveru a emailové adresy pro administrátora. Jelikož tento server bude běžet lokálně a nebude přístupný z Internetu, je možné zvolit libovolná pojmenování, nejlépe však smyšlená k ochraně vlastní identity.




**Obr. 3.8. Nastavení informací o serveru**

Po nainstalování je možné ověřit funkčnost serveru v dolní systémové liště a zadáním adresy http://localhost/ do internetového prohlížeče.

Následně lze přistoupit k samotné konfiguraci skryté služby. Nejdříve je nutné modifikovat konfigurační soubor „torrc“, který se nachází v případě nainstalované verze zde:

*C:\Users\Administrator\AppData\Local\Vidalia\torrc*

Editujeme tento soubor a přidáme následující kód pro nastavení skryté služby:

```
HiddenServiceDir C:\HiddenServiceSettings/
HiddenServicePort 80 127.0.0.1:80
HiddenServicePort 443 127.0.0.1:443
```

Pomocí výše zmíněných příkazů je nastavena složka, do které Tor uloží informace o skryté službě. Je nutné, aby tato složka nebyla uložena ve stejné složce jako nainstalovaný webový server z důvodu zabezpečení.

Druhá položka specifikuje virtuální port, na kterém bude služba dostupná a adresu přesměrovávané IP adresy a portu. V našem případě localhost na portu 80. Poslední příkaz slouží pro přístup na server pomocí zabezpečeného spojení HTTPS fungující na portu 443.

Následně soubor uložíme a restartujeme Vidalii. Po opětovném spuštění služby Tor jsou vygenerovány dva soubory „hostname“ a „private_key“. Soubor „hostname“ obsahuje Tor adresu serveru „wnv3zrtcklmlkprx.onion“, pomocí niž je skrytá služba dostupná. Tato adresa je určena k šíření a informování uživatelů. Druhou vygenerovanou položkou je soubor „private_key“, který obsahuje privátní RSA klíč pro OP k šifrování celé komunikace. V případě kompromitace tohoto klíče bude možné kompromitovat skrytou službu. Příklad privátního klíče:


```
-----BEGIN RSA PRIVATE KEY-----
MIICXAIBAAKBgQDAjR3jiYVKU9cRIb2a9H2hLwGxXKK6l9LLi2TrnCOuQPQWzaDA
u3VpoB+Pem101S32+rRbNojnMg31n7xVsJwdXW9Ee2YRrSojMDRqFOUaYNaqrkYT
65zXumoO0avIEiGYAgNVtQmFU7NhvD+9X5SgUI7Nwd0qsXoXJ/goXt90hwIDAQAB
AoGAbEgvkcDJY1YosOqvjM+HQriDlDFLFfHBSlP2igUBj8HZQmWazCffTGoOtYoc
ucIUmjq8aDsP3Cwq64FL0UiDUYo2AIW9ZFWiKweA8pF2cD6tb+LD/EO7njnKw6bZ
2aZ6bZPpjlbVArQ1vrEswgTboNl8nr5Ccxt0uXYD94j7P5ECQQDo2Ju2HO0cKejv
UX03ig1COQTSbChBV6kmVo6cM9wyI9FvTxDCF5PcQUFMDPwlxoaMq1q5ICmtu1Id
7k+YnyzJAkEA07LAVXCG5VLTkinS8h4dYvtgDHYTl+7pQpH64V/+0l6QFAbH4Yyt
vXzRR9aLHjerjMudlIAQG/exoi1C5d1OzwJAMA9dfGZWDoUb/iGMveTxvaR1OVV3
rZ0BXddp0hXmMjcwaFAifl1NeJ3Dc4YOmWFzrh9EN4fK8oACgAkLRs47+QJAVjES
I/dDxPy7DWif7VobhuBoE8rDYXmxWRxMdBR8kLRVnwG/wfJgDqVSJBiS3cO2wgVo
7DxT0r8/gJNDCInZewJBALLlH3RUxxmgjDaSPMquE1DRm8TjVbUSidsdsgJkxbqO
frYYBVnn7xlL5rAhqX3MshsNd0GX0v7vva0phYng2Y0=
-----END RSA PRIVATE KEY-----
```


## 3.4. Implementace zabezpečeného anonymního přístupu pomocí HTTPS

Doposud byl server se Skrytou službou nastaven tak, aby byla šifrována komunikace mezi OP a OR. Tím bylo docíleno zabezpečeného přenosu v rámci sítě Tor. Takto zabezpečené spojení však není dostatečné, jelikož komunikace od hraničních OR není šifrována a využívá standardní protokol HTTP.

### Jednosměrná SSL autentifikace

Jako nejlepší řešení pro klasický anonymní a zabezpečený přístup uživatele vůči serveru je použití protokolu HTTPS s jednosměrnou SSL autentifikací. Díky tomu je možné docílit koncového zabezpečení mezi klientem a server, ve kterém se server identifikuje vůči klientovi svým certifikátem.

Toto řešení však obsahuje problém s důvěryhodností certifikátu, který bude použit serverem. Při běžné implementaci certifikátu je nutné, aby tento certifikát ověřila třetí strana, která bude ručit za správnost a autenticitu osoby, která tímto certifikátem disponuje. To však v tomto případě není možné, jelikož by bylo nutné Certifikační Autoritě (CA) sdělit veškeré informace o naší osobě (jméno, adresa, email atd.), které by CA následně porovnala s veřejně dostupnými údaji a požadovala ověření na udané korespondenční adrese. Tím by došlo ke kompromitování našeho serveru, na kterém běží skrytá služba.

Částečným řešením tohoto problému je vytvoření vlastní certifikační autority, která se postará o podepsání certifikátu, avšak je nutné, aby tuto certifikační autoritu taktéž uznal klient a vybudoval si k ní důvěru. I přes to, že nejsme schopni plně ověřit vlastníka, zůstávají zde výhody šifrování a zamezení možného přečtení přenášených zpráv.

Pro vytvoření a konfiguraci certifikátu, klíčů i certifikační autority, je vhodné použít doplněk OpenSSL, který je součástí instalace Apache.

## Vytvoření vlastní certifikační autority

Vytvoření vlastní certifikační autority je téměř totožné s vytvořením certifikátu samotného. Důležitou částí zabezpečení je použití 2048 bitového RSA klíče, který není aktuálně možné prolomit. Využití doposud používaného 1024 bitového klíče není doporučeno, jelikož se blíží konec jeho životnosti, spočívající v jeho neprolomitelnosti. Sebemenší náznak tohoto prolomení by měl za následek nedůvěru ke všem těmto klíčům. Proto je od ledna 2014 doporučeno všem prohlížečům a stránkám využívat minimálně 2048 bitů dlouhého RSA klíče. Postup vytvoření je následující:

1. Nejdříve je nutné přesunout se do složky s programem OpenSSL, který pracuje skrze příkazový řádek [7].

   ```
   cd "C:\Apache2.4\bin"
   ```

2. Následně je nutné upravit hodnotu systémové proměnné Windows OPENSSL_CONF tak, aby odkazovala na umístění souboru openssl.cnf nebo lze použít parametr -config openssl.cnf, který specifikuje cestu k předdefinovanému nastavení OpenSSL:

   ```
   set OPENSSL_CONF=C:\Apache2.4\conf\openssl.cnf
   ```

3. Nyní můžeme vytvořit privátní klíč RSA s blokovou šifrou Triple DES a délkou klíče 2048 bitů.

   ```
   openssl genrsa -des3 -passout pass:cR3Ns2dPteS4RF5
   -out  CA/key/CA.key 2048

   Loading 'screen' into random state - done
   Generating RSA private key, 2048 bit long modulus
   ..................................................................
   ....+++
   e is 65537 (0x10001)
   ```

4. Dalším krokem je vytvoření souboru CSR ( Certificate Signing Request) určený pro CA nebo pro vlastnoruční podpis certifikátu. V souboru CSR jsou specifikovány informace o vlastníkovi. V našem případě zadáváme informace o naší certifikační autoritě. Vyplnění těchto hodnot je zcela volitelné, ale mělo by být vyplněno tak, aby byla zachována anonymita serveru, ale zároveň byly poskytnuty informace pro identifikaci.

```
openssl req –new  -key CA/key/CA.key -out
CA/request/requestCA.cbr -config openssl.cnf

Enter pass phrase for CA/key/CA.key:
Loading 'screen' into random state - done
You are about to be asked to enter information that will be
incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished
Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:XX
State or Province Name (full name) [Some-State]:TOR Network
Locality Name (eg, city) []:Tor
Organization Name (eg, company) [Internet Widgits Pty
Ltd]:wnv3zrtcklmlkprx.onion
Organizational Unit Name (eg, section) []:wnv3zrtcklmlkprx.onion
Common Name (e.g. server FQDN or YOUR name)
[]:wnv3zrtcklmlkprx.onion
Email Address []:.

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:.
An optional company name []:.
```

5. Následuje poslední krok, při kterém podepíšeme vlastní požadavek na certifikát. Tím dostaneme vlastnoručně ověřený certifikát, který bude představovat naši certifikační autoritu.

```
openssl x509 –req  -days 3650 -in CA/request/requestCA.cbr -out
CA/certificate/CAcert.crt -signkey CA/key/CA.key

Loading 'screen' into random state - done
Signature ok
subject=/C=XX/ST=TOR
Network/L=Tor/O=wnv3zrtcklmlkprx.onion/OU=wnv3zrtcklmlkprx.
onion/CN=wnv3zrtcklmlkprx.onion
Getting Private key
Enter pass phrase for CA/key/CA.key:cR3Ns2dPteS4RF5
```

## Postup vytvoření certifikátu pro server:

Vytvoření certifikátu je téměř totožné s vytvořením certifikátu, potažmo certifikační autority.

1. Vytvoříme privátní klíč serveru:

```
openssl genrsa -des3 -passout pass:4VSGjBtrLGeQkEk
-out  server/key/server.key 2048

Loading 'screen' into random state - done
Generating RSA private key, 2048 bit long modulus
.....................................+++
...+++
e is 65537 (0x10001)
```

2. Vytvoříme požadavek pro certifikační autoritu. Nejdůležitějším polem při vyplňování je CN (Common Name), které představuje plně kvalifikované doménové jméno. V našem případě se jedná o adresu Skryté služby wnv3zrtcklmlkprx.onion.

```
openssl req -new -key server/key/server.key -out
server/request/requestserver.cbr -config openssl.cnf

Enter pass phrase for server/key/server.key:
Loading 'screen' into random state - done
You are about to be asked to enter information that will be
incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished
Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:XX
State or Province Name (full name) [Some-State]:Tor Network
Locality Name (eg, city) []:Tor
Organization Name (eg, company) [Internet Widgits Pty
Ltd]:wnv3zrtcklmlkprx.onion

Organizational Unit Name (eg, section) []:wnv3zrtcklmlkprx.onion
Common Name (e.g. server FQDN or YOUR name)
[]:wnv3zrtcklmlkprx.onion
Email Address []:.

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:PLPP5x9zn59ZSLC
An optional company name []:wnv3zrtcklmlkprx.onion
```

3. Následně je nutné tento požadavek podepsat. Certifikát nebude vlastnoručně podepsán jako v předchozím případě, ale bude podepsán certifikátem a privátním klíčem certifikační autority stejným způsobem, jako by jej podepsala ověřená CA.

```
openssl x509 -req -days 3650 -in server/request/requestserver.cbr
-out server/certificate/Server
cert.crt -CAkey CA/key/CA.key -CA CA/certificate/CAcert.crt –
Cacreateserial

Loading 'screen' into random state - done
Signature ok
subject=/C=XX/ST=Tor
Network/L=Tor/O=wnv3zrtcklmlkprx.onion/OU=wnv3zrtcklmlkprx.o
nion/CN=wnv3zrtcklmlkprx.onion
Getting Private key
Enter pass phrase for CA/key/CA.key:cR3Ns2dPteS4RF5
```

Nyní máme vygenerované všechny certifikáty, které obsahují veřejné klíče, a privátní klíče, nutné k dešifrování komunikace.

Vytvořený privátní klíč bude při každém startu serveru vyžadovat zadání nastaveného hesla. To není žádoucí v případě, že nemáme stálý přístup k serveru v případě pádu aplikace či restartu serveru. Jedním z možných řešení je odstranění hesla z tohoto klíče. To způsobí, že server nebude vyžadovat zadání hesla, avšak klíč bude uložen v dešifrovaném stavu. Proto je nutné na serveru nastavit omezení přístupu, aby nebylo možné volně číst nechráněný klíč.

```
copy server.key server.key.org
openssl rsa –in server/key/server.key.org -out
server/key/server.key

Enter pass phrase for server.key.org:
writing RSA key
```

Nově vytvořené soubory „server.key“ a „Servercert.crt“ přesuneme do nové složky *\Apache24\conf\ssl*. Následně je nutné nastavit server Apache tak, aby používal zabezpečené spojení za pomoci nově vytvořeného certifikátu a privátního klíče.

Toho dosáhneme úpravou hlavního konfiguračního souboru „httpd.conf“ a načtením modulu:

```
LoadModule ssl_module modules/mod_ssl.so
```

A následnou úpravou souboru *„\Apache24\conf\extra\httpd-ssl.conf“*, kde nastavíme odposlech na portu 443.

```
Listen 443
```

Apache může pracovat na různých portech, IP adresách a umožňuje rozlišovat i doménová jména. Díky tomu může obsluhovat více webů zároveň. Nastavením

virtuálního hosta můžeme specifikovat nastavení pro jednotlivé weby. Ten je definován:

```
<VirtualHost *:443>
        ServerName wnv3zrtcklmlkprx.onion
        ServerAlias wnv3zrtcklmlkprx.onion
</VirtualHost>
```

Dále je nutné zapnout samotný SSL modul pro virtuální hosta:

```
SSLEngine on
```

Taktéž je vhodné specifikovat, jaké protokoly jsou podporovány pro komunikaci. Následujícím příkazem povolíme všechny verze a zamezíme připojení skrze SSL:

```
SSLProtocol all -SSLv2 -SSlv3
```

Příkazem `SSLCipherSuite` můžeme specifikovat šifry které budou podporovány pro komunikaci. Díky tomuto je možné zvolit šifry pro lepší zabezpečení nebo naopak pro co nejrychlejší šifrování a tím snížit výpočetní zátěž na server.

Taktéž implementujeme cestu k certifikátu serveru a privátnímu klíči:

```
SSLCertificateFile "C:/Apache24/conf/ssl/Servercert.crt"

SSLCertificateKeyFile "C:/Apache24/conf/ssl/server.key"
```

Správné ověření konfigurace je možné po restartu serveru ověřit zadáním adresy https://localhost nebo https://wnv3zrtcklmlkprx.onion na klientském Tor prohlížeči.

Následně je vhodné upravit chování serveru Apache tak, aby při přístupu na adresu wnv3zrtcklmlkprx.onion bylo toto spojení přesměrováno na zabezpečené využívající HTTPS.

Toho lze docílit mnoha způsoby. Nejvhodnějším je přidáním příkazů do httpd konfiguračního příkazu.

```
RewriteEngine on
ReWriteCond %{SERVER_PORT} !^443$
RewriteRule ^/(.*) https://%{HTTP_HOST}/$1 [NC,R=301,L]
```

## Obousměrná SSL autentifikace

Doposud jsme používali nejrozšířenější způsob, jakým se zabezpečuje komunikace mezi klientem a serverem. Tím a šifrovanou komunikací samotného Toru je dosaženo poměrně vysokého zabezpečení a i anonymity, jelikož není nikde veřejně znám a evidován žádný DNS záznam pro náš server a tedy uživatel přistupující na náš server jej mohl získat nějakou jinou například námi kontrolovanou cestou. Zároveň jsme v certifikátu nezmínili žádné kompromitující informace a vytvořili jej tak, aby obsahoval co nejméně informací, ale i částečnou identifikaci.

To vše však neřeší další problém a to je autentifikace anonymního uživatele. Jak již bylo popsáno docílit autentifikace bez znalosti uživatele a povolit mu přístup na server je velmi obtížné.

Jako řešení je možné použít certifikát i pro klienta. Díky němu budeme moci identifikovat uživatele a budeme mít přehled o tom, kdo jej vlastní, jelikož jej sami vydáme.

Postup vytvoření certifikátu pro uživatele je totožný s vytvořením certifikátu pro server:

```
openssl genrsa -des3 -passout pass:g7Tj3pgchypnDuK -out
user/key/client.key 2048
```

```
openssl req -new -key user/key/client.key -out
user/request/requser.cbr
```

Výše zmíněné příkazy nemusí být vytvářeny a vyplňovány námi, ale klient je také může sám vytvořit a následně je nechat podepsat. Následně stačí požadavek na certifikát podepsat naši vlastní CA.

```
openssl x509 -req -days 3650 –in user/request/requser.cbr -out
user/certificate/certuser.crt –CA key CA/key/CA.key -CA
CA/certificate/CAcert.crt –Cacreateserial
```

Nyní jsme vydali certifikát pro klienta. Je tedy nutné nastavit server tak, aby akceptoval právě a pouze námi vydané certifikáty. K tomu nám poslouží příkaz do nastavení httpd-ssl.conf, kterým přidáme cestu k CA, která bude ověřovat klientské certifikáty.

```
SSLCACertificateFile "C:/Apache24/conf/ssl/CAcert.crt"
```

Server tedy ověřuje platnost certifikátu a to ve výchozím nastavení do hloubky deset. To znamená, že server se dotáže CA na podepsaný certifikát a ten je vystaven jednou z organizací spadajících pod hlavní CA. V našem případě by tento certifikát byl nedůvěryhodný, protože nemáme CA ověřenou a není ji důvěřováno. Je tedy nutné nastavit hloubku na jedna. Tím bude docíleno, že certifikát bude uznán, pokud je podepsán CA přímo známou serveru. Toho jsme docílili předchozím nastavením. Nastavením na hloubky na nula, bychom akceptovali pouze vlastnoručně podepsané certifikáty klientů, což může být taktéž vhodné pro anonymní přístup.

```
SSLVerifyDepth  1
```

Následující příkaz vynutí požadavek na certifikát klienta při přístupu na server.

```
SSLVerifyClient require
```

## Nastavení privátního přístupu na server

V případě, že se klient zdárně autentizuje, je mu umožněn přístup na server. Můžeme využít vlastností modulu SSL a filtrovat přístup v závislosti na jednotlivých parametrech klientova certifikátu například v podobě jména, státu, verze použitého certifikátu atd. Tento způsob autentifikace může plně nahradit přihlašovací údaje v podobě uživatelského jména a hesla. Nastavení omezeného přístupu na složky může být proveden následovně:

```
<Location /US2>
    SSLVerifyClient require
    SSLVerifyDepth  1
    SSLRequire %{SSL_CLIENT_S_DN_CN} eq "USER2" and
%{SSL_CLIENT_M_SERIAL} eq "DDBC2738112CAE15"
</Location>
```

Tímto nastavením omezíme přístup na uživatele s doménovým jménem nebo uživatelským na „USER2“ a ověříme klientovo sériové číslo.

## Kruhový podpis

I přes to, že byla implementována obousměrná SSL autentifikace, bylo dosaženo zabezpečení komunikace společně s ověřením uživatelů, nebylo prozatím dosaženo dostatečné anonymity.

Tento problém pomohou vyřešit kruhové podpisy, které v kryptografii slouží k podpisu zpráv skupinou uživatelů držící privátní klíč. Není tedy jednoduše možné determinovat, který uživatel zprávu podepsal, ale víme, že je to ověřený uživatel, jelikož vlastní privátní klíč.

Jednoduchým příkladem může být dokument, který někdo orazítkuje firemním razítkem. Nevíme kdo to byl, ale víme, že razítko vlastní jen určitý okruh lidí, kteří jej mají uschované. Tudíž můžeme důvěřovat tomuto dokumentu, že byl vydán důvěryhodnou osobou.

Tento princip lze přenést i na uživatelské certifikáty. Poměrně jednoduše lze jeden certifikát předat vícero uživatelům a vydávat jej za skupinový. Abychom docílili většího zabezpečení a v případě nutnosti, klientovy identifikace, využijeme kryptografický kontejner PKCS 12.

Kryptografický kontejner PKCS 12 slouží k uložení mnoha kryptografických objektů v podobě certifikátů, privátních, veřejných klíčů a jiných. Díky tomuto kontejneru je možné přenést všechny potřebné soubory a zároveň je jednoduše implementovat do webového prohlížeče.

Další výhodou je, že kontejner může být zašifrován jednorázovým heslem, který zamezí přístupu neověřených osob k celému obsahu kontejneru. Tento princip může být použit k vydání skupinového certifikátu s privátním klíčem mnoha osobám, ale jednotlivé

balíčky budou zabezpečeny různými hesly. Tím bude dosaženo toho, že uživatelé přistupující na server skupinovým certifikátem a ověření vlastnictvím privátního klíče budou anonymní, jelikož nebude možné zjistit, kdo daný certifikát používá.

V případě diskreditace tohoto kruhového certifikátu tedy není problém vystavit nový, zašifrovat jej jednorázovými hesly a vydat je příslušným osobám.

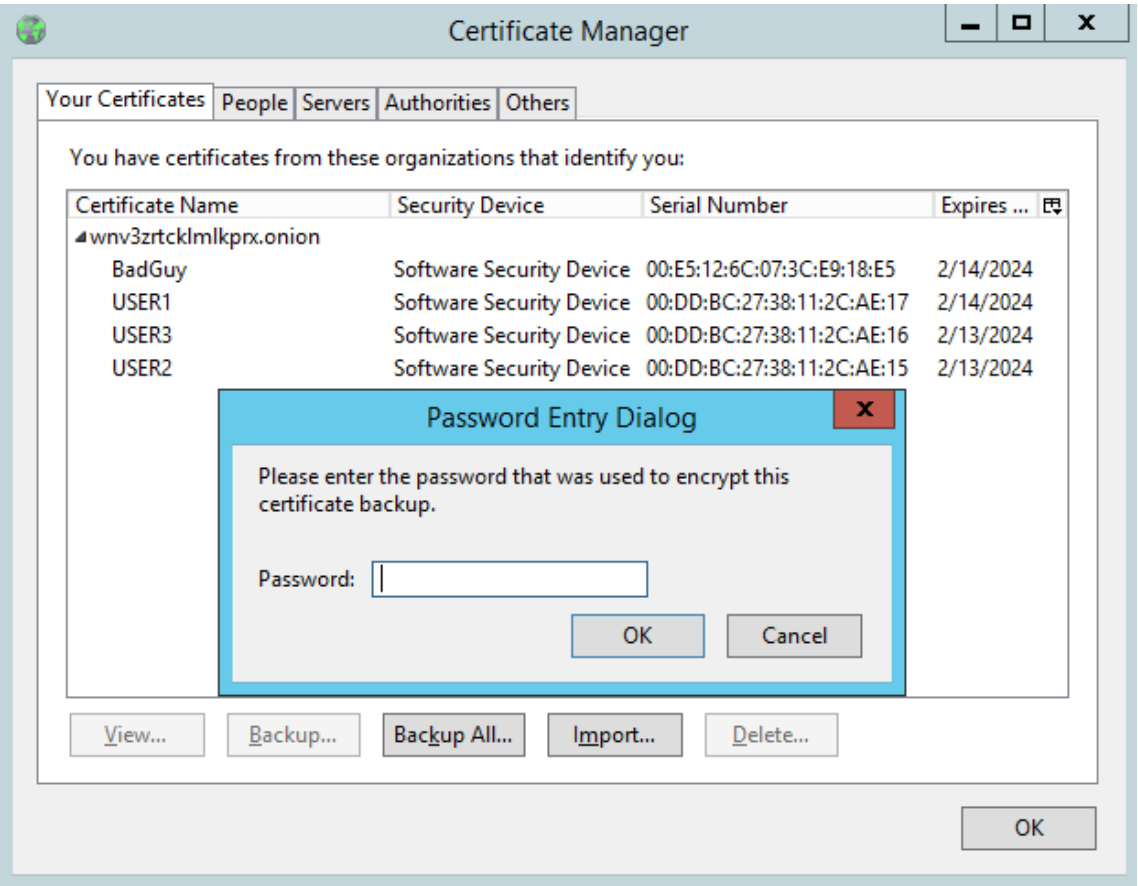
Postup vytvoření kontejneru je následující, do kontejneru bude nahrán certifikát klienta, jeho privátní klíč a dále je možné přidat i certifikát CA, který může být použit k vytvoření důvěry mezi prohlížečem a serverem.

```
openssl pkcs12 -export -clcerts -in user/certificate/certuser.CRT
-certfile CA/certificate/CAcert.crt -inkey user/key/client.KEY
-out user/certificate/certuser.P12

Loading 'screen' into random state - done
Enter pass phrase for user/key/client.KEY: g7Tj3pgchypnDuK
Enter Export Password:ZLzcb4cafm7md4dr
Verifying - Enter Export Password:CP5rCMUkeWaebdjc
```

*Implementace do prohlížeče Toru:*

Menu => Option => Advanced => Certificates => View Certificates

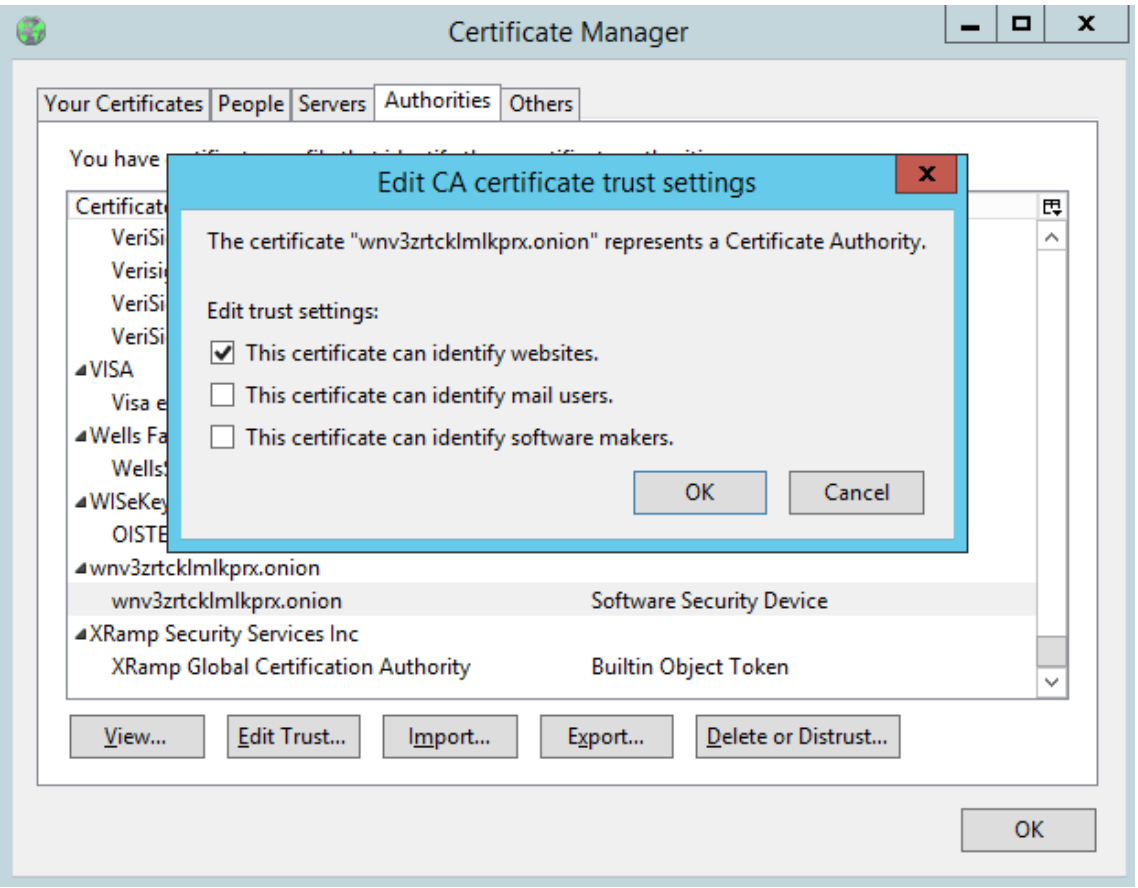



**Obr. 3.9. Importování kontejneru do prohlížeče Toru**

Tímto se dostaneme do nastavení manažera certifikátů, kde tlačítko import slouží pro nahrání vytvořeného kontejneru zabezpečeného heslem. Následně prohlížeč obsahuje vše nutné pro přístup na náš server. Jak je patrné, požadavky na klienta jsou minimální. Pro přístup stačí Tor klient, adresa serveru a vlastnictví certifikátu a hesla k němu.

Při pokusu o přístup bude prohlížeč oznamovat, že spojení se serverem je nedůvěryhodné. To je způsobeno tím, že prohlížeč nebyl schopen ověřit totožnost certifikátu u jedné z důvěryhodných CA. Buď je možné vždy při přístupu potvrdit, že chápeme rizika a vstoupit nebo můžeme využít námi vytvořeného kontejneru, který obsahuje i naši CA.

Pro nastavení důvěry naší CA pro ověřování webových stránek otevřeme opět manažera certifikátů prohlížeče Firefox, přepneme se na záložku Authorities. Zde jsou všechny CA, kterým prohlížeč důvěřuje. Díky předchozímu importu kontejneru je zde i certifikát wnv3zrtcklmlkprx.onion, u kterého editací důvěry nastavíme možnost identifikovat webové stránky.




**Obr. 3.10. Nastavení důvěry pro kořenovou CA**

*HTTPS MitM*

Za předpokladu že by datový provoz byl směřován přes jeden ze zkažených výstupních uzlů a byl by podvržen certifikát klientovi, má Firefox zabudovanou ochranu. Tato ochrana se aktivuje v případě použití neověřeného vlastnoručně podepsaného

certifikátu nejdříve oznámením o nedůvěryhodnosti, viz výše a dále opětovným znovunačtením certifikátu v případě, že uživatel přidá certifikát do výjimek.

Při použití tohoto útoku při každém *n* pokusu, bude možná úspěšnost tohoto útoku $n^2$.

## Autorizace uživatelů přistupujících na HS

V posledních době byly uveřejněny útoky, které byly schopny a některé stále mohou ohrozit dostupnost a hlavně identitu HS.

Z tohoto důvodu je vhodné zvýšit zabezpečení pomocí autentizace uživatelů při získávání IP pro HS.

Jak bylo popsáno v kapitole věnující se útokům na HS, existují dva protokoly pro zabezpečení a to základní a neviditelný.

Základní ochrana je implementována editací souboru torcc na straně HS a to přidáním následujícího příkazu:

```
HiddenServiceAuthorizeClient basic Client1
```

Hodnota Client1 určuje libovolný název, který může obsahovat až 16 znaků. Po uložení souboru a spuštění Toru, bude vygenerováno nové adresa HS.

```
wnv3zrtcklmlkprx.onion sGrwecl1lmSMGda/0zVtsw
```

Jak je zřejmé, adresa HS zůstává stejná, přidává se pouze 22 znaků dlouhý descriptor cookie, který je následně nutné implementovat do konfiguračního souboru torcc i na straně klienta. Lokace souboru je .\Tor Browser\Data\Tor\torcc. Následně je nutné přidat záznam ve tvaru:

```
HidServAuth wnv3zrtcklmlkprx.onion sGrwecl1lmSMGda/0zVtsw
```

V případě neviditelné autorizace je klientská část totožná, mění se pouze příkaz na straně serveru a to na:

```
HiddenServiceAuthorizeClient stealth Client1, … Client16
```

Po spuštění jsou všechny adresy uloženy ve složce s privátním klíčem a adresou HS v souboru hostname:

```
t6npvgudjgfih72a.onion sGrwecl1lmSMGda/0zVtsx # client: Client1
jezz3artjhgfpjlg.onion 3gERAptfFSLyY7xGPnaHHB # client: Client2
o56jzoecuflzebtx.onion i521weqB/OuDJtww6G4lsB # client: Client3
wcp4xcigdnrwsydk.onion 6NJSsOrCCmzr61gWAi2EWB # client: Client4
a5axsz45zhwtt2dm.onion 4IVKhlPcn/eAzG3zwWVRYR # client: Client5
vqf5guxwhmuaousc.onion jOYhwE26IeSujtelb4KjRh # client: Client6
p3bb7f4e3lv5uxxh.onion CPgaM5DO3FwcToCu96Zi6x # client: Client7
tbxyxi2tgsp6daol.onion DzWABxErpQmAs/Y3nRUbLR # client: Client8
pmsqhia7efviypnd.onion lxDxFYo4K9VY1RWOydGHJx # client: Client9
nrysdhpcoyuwferg.onion 6TYh2UOTVqL2LMKfOdfw1h # client: Client10
fmfi2bts5krn2dyq.onion biFwS7mOyfAS8Vsnn089Ch # client: Client11
thpelrcdvakaozdg.onion D/tu/+5ku3uxYmEAtPXNsB # client: Client12
xdh2rmq3ztlkdwzl.onion hseHvSDcsVlqV+3nhVaMjB # client: Client13
hcwdksti4hawzvj6.onion Ggh3DrRIveOJKjxUbO7YYx # client: Client14
z537rqozdzmiahhj.onion 2c5TCNXlurGNorEnRCryjB # client: Client15
6whqlzpwwelieqbr.onion wPx1YlESkvi/OYjHjmlBSR # client: Client16
```

Soubor client_keys obsahuje všechny privátní RSA klíče pro jednotlivé klienty a jejich descriptor cookie.

V případě použití více adres je nutné vygenerovat nové klíče a certifikáty pro jednotlivé adresy a podepsat je naší certifikační autoritou. Následně implementovat na server Apache přidáním dalších virtuálních portů. Každý webový server by měl mít dedikovanou IP adresu a port pro každou zmíněnou stránku, aby byla zajištěna dostatečná kapacita a dostupnost služby. Je také možné využít jednu IP adresu a port. V případě využití certifikátů však může dojít k problémům a ztrátě spojení. Ty jsou spojeny s tím, že Apache neví jaké doménové jméno je požadováno do té doby, dokud nenaváže SSL spojení s klientem. Díky tomu může dostat klient špatný certifikát a spojení může být zahozeno nebo označeno za nebezpečné.

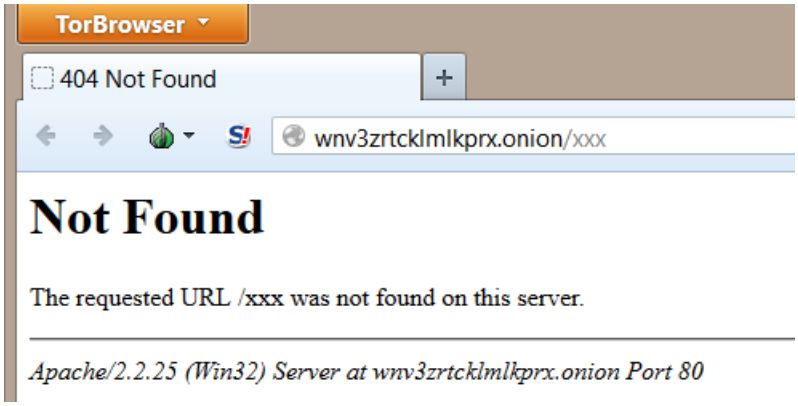
## Úprava hlavičky odpovědi HTTP

Uživatel nebo útočník, který zjistí adresu skrytého serveru a přistoupí na něj, může zjistit několik zajímavých informací, které mohou posloužit ke zjištění možných útoků vůči serveru. Tyto informace se vyskytují v odpovědních zprávách na požadavky „http get“ a u chybových oznámení.

```
Date: Sat, 02 Nov 2013 12:40:52 GMT
Server: Apache/2.2.25 (Win32)
Content-Length: 289
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=iso-8859-1

200 OK
```

A také v podpisu neexistující stránky.




**Obr. 3.11. Informace o serveru a platformě OS**

Záhlaví této zprávy se může lišit v závislosti na použitém webovém serveru, zde je zmíněna hlavička nejrozšířenějšího serveru Apache, z nichž je patrný název serveru, hlavní verze, vedlejší verze, druh platformy operačního systému a hodnoty časovačů. Tyto informace sami o sobě nejsou nijak závadné, avšak poskytují útočníkovi informace, pomoci nichž je schopen cíleně útočit na skrytou službu, využívaje jejich trhlin.

K úpravě této hlavičky je nutné upravit konfigurační soubor „httpd.conf“ přidáním příkazu:

```
ServerTokens ProductOnly
ServerSignature Off
```

Výsledkem je zpráva:

```
Date: Sat, 02 Nov 2013 13:42:21 GMT
Server: Apache
Last-Modified: Sat, 02 Nov 2013 09:51:56 GMT
Etag: "10000000153a6-38-4ea2ea22a5542"
Accept-Ranges: bytes
Content-Length: 56
Content-Type: text/html

200 OK
```

Zmíněné příkazy vypnou informace o verzi serveru a platformě operačního serveru. Položka typu serveru zůstala ponechána, jelikož po změně by bylo stále možné z chování serveru určit, že se jedná právě o Apache server. V případě pokusu o zmatení útočníka změnou tohoto jména je nutné extrahovat instalační balíček MSI, upravit soubor „ap_release.h“ a opět sestavit balíček msi.

```
#define AP_SERVER_BASEVENDOR "Apache Software Foundation"
#define AP_SERVER_BASEPROJECT "Apache HTTP Server"
#define AP_SERVER_BASEPRODUCT "Apache"

#define AP_SERVER_MAJORVERSION_NUMBER 2
#define AP_SERVER_MINORVERSION_NUMBER 2
#define AP_SERVER_PATCHLEVEL_NUMBER   25
#define AP_SERVER_DEVBUILD_BOOLEAN    0
```

Hodnoty verze není vhodné upravovat, protože jsou na ně navázány rozšiřující moduly a zároveň již nejsou zobrazovány v hlavičkách.

Nově zobrazené položky „Last-Modified“ a „Etag“ slouží jako otisky obsahu. V případě, že statická stránka nebyla změněna od poslední návštěvy zašle server tuto hlavičku a prohlížeč použije již předem načtenou stránku ze svého zásobníku.

## CGI skript

Při používání webového serveru může být nutná jistá interakce a s tím spojený dynamický obsah stránek. Jelikož je prohlížeč Toru nastaven tak, aby nepoužíval javascript, musí se vytváření dynamického obsahu přesunout na stranu serveru.

K tomuto účelu slouží CGI skript (Common Gateway Interface), který umožňuje spustit na serveru program, který vygeneruje například html stránku s požadovaným obsahem.

Základním programovacím jazykem je Perl nebo může být využito C, C++ případně Javy.

Nutnými prerekvizitami pro správnou funkčnost je implementace knihoven pro daný jazyk. V našem případě byl zvolen Perl, který je dostupný z [23].

Následně je nutné v httpd.conf povolit modul pro spustitelné programy.

```
LoadModule cgi_module modules/mod_cgi.so
```

Aby bylo možné spouštět programy mimo výchozí CGI složku, je nutné v cílové lokaci povolit spouštění CGI souborů s koncovkou .pl:

```
AddHandler cgi-script .pl
Options +ExecCGI
```

Jelikož využíváme zabezpečené spojení, je taktéž nutné povolit CGI proměnné pro SSL spojení.

```
<FilesMatch "\.(pl|cgi|shtml|phtml|php)$">
    SSLOptions +StdEnvVars
</FilesMatch>
```

Výsledek takovéhoto skriptu může být například výpis informací z klientova certifikátu.

**Informace o vašem úctu:**

| | |
|---|---|
| **Uživatelské jméno:** | USER2 |
| **Lokace:** | Europe |
| **Sériové číslo certifikátu:** | DDBC2738112CAE15 |
| **Certifikát platný od:** | Feb 15 16:33:50 2014 GMT |
| **Certifikát platný do:** | Feb 13 16:33:50 2024 GMT |
| **Vydáno CA:** | wnv3zrtcklmlkprx.onion |

**Obr. 3.12. Výpis CGI skriptu**

## 3.5. Výsledky měření

Pro ověření přenosových rychlostí jednotlivých linek byly použity jednotlivé předpřipravené soubory o přesně dané velikosti. Následně byla provedena měření přímého propojení, spojení skrze VPN s výstupními body v Brně (Česká republika), Corku (Irsko), Austinu (USA). Poslední měření proběhlo skrze veřejnou proxy umístěnou v Ohiu (USA). Byla provedena dvě měření s odstupem jednoho týdne ve stejný čas pro ověření rychlostí a stability, avšak druhé měření pro proxy server, umístěný v USA, nebylo možné provést, jelikož daný server ukončil svoji činnost a stal se tak nedostupným pro další měření.

Ze zaznamenaných průběhu je patrné (viz příloha), že nejlepší stabilita a přenosová rychlost byla dosažena skrze přímé spojení. To je způsobeno nejkratší přenosovou vzdáleností a pouze jedním zařízením v přenosové cestě.

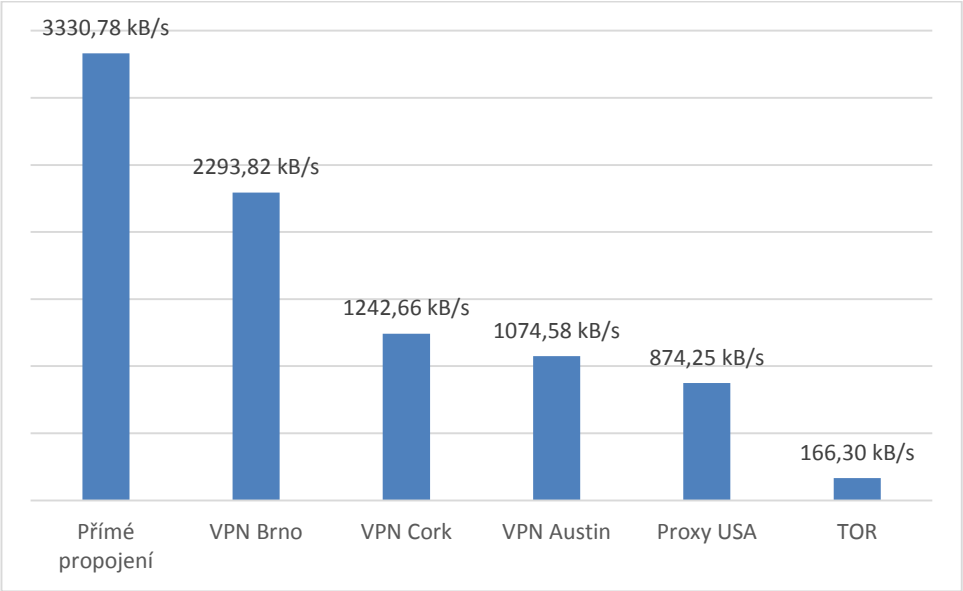
Další měření věnující se VPN ukázala, že přenosové rychlosti a stabilita jsou velmi náchylné na vzdálenost zdroje a cíle komunikace. Data necestují nejkratší možnou cestou, jako v případě přímého propojení, ale jsou směrována skrze VPN tunel k výstupnímu bodu, který komunikuje s cílovou destinací. Přenos dat zpět funguje stejným způsobem v opačném směrů, tady data jsou poslána k bráně VPN, která je zašle zpět výchozí stanici. To je dosti neefektivní a ve výsledku je patrný dosti podstatný pokles přenosové rychlosti. Ten je umocněn vzdáleností brány a výchozí stanice. Rozdíl mezi bránou v Brně a v Austinu činí až dvojnásobek přenosové rychlosti.

**Tab. 3.1. Výsledné hodnoty průměrných rychlostí v závislosti na druhu spojení a velikosti souborů**

| Velikost | Přímé propojení | VPN Brno | VPN Cork | VPN Austin | Proxy USA | TOR |
|---|---|---|---|---|---|---|
| **MB** | kB/s | kB/s | kB/s | kB/s | kB/s | kB/s |
| **10** | 1824.24 | 1561.99 | 1094.54 | 218.97 | 783.27 | 101.18 |
| | 1567.97 | 1419.08 | 1090.46 | 254.90 | | 159.26 |
| **20** | 2439.28 | 1563.69 | 1792.61 | 1217.49 | 1093.33 | 91.04 |
| | 3161.83 | 2225.40 | 1247.77 | 1124.90 | | 202.79 |
| **30** | 2715.90 | 1781.02 | 1476.93 | 935.58 | 937.54 | 114.00 |
| | 3648.10 | 2626.29 | 821.17 | 586.69 | | 181.22 |
| **40** | 3136.18 | 2086.72 | 1936.36 | 1910.87 | 1043.39 | 157.32 |
| | 3917.33 | 2431.50 | 811.19 | 608.33 | | 233.07 |
| **50** | 3792.83 | 2530.35 | 1076.45 | 1658.51 | 1115.39 | 105.27 |
| | 3690.03 | 2283.75 | 960.41 | 1989.96 | | 251.95 |
| **100** | 3628.00 | 2614.47 | 1434.80 | 889.44 | 579.78 | 126.35 |
| | 3789.90 | 2880.64 | 1480.57 | 1023.75 | | 240.22 |
| **200** | 4581.44 | 2741.16 | 1105.93 | 1364.52 | 567.06 | 175.98 |
| | 4737.89 | 3367.36 | 1068.10 | 1260.29 | | 188.63 |
| **Průměr** | **3330.78** | **2293.82** | **1242.66** | **1074.58** | **874.25** | **166.30** |

Z průběhů přenášených dat skrze proxy server je patrná nižší přenosová rychlost vůči VPN umístěné v Austinu. To je především způsobeno větším vyhrazením datových prostředků mezi výchozí stanicí a bránou VPN, jelikož komunikace skrze proxy server je směrována společně s běžným datovým provozem.

Přenos dat srze proxy server může být dále různě omezen a to tak, aby bylo vyhověno velkému množství komunikujících uživatelů. Při testování spojení bylo zpozorováno, že data přenášená mezi proxy serverem a výchozí stanicí nejsou nijak šifrována a to i přes vynucené šifrované spojení na straně cílového serveru. To představuje velmi zásadní problém v podobě možného odchytávání všech přenášených dat, které jsou volně čitelné jak ISP, tak případnému útočníkovi. V tomto případě tak nejsou data chráněna celou cestu mezi USA a Brnem.




**Graf 3.1. Závislosti průměrné rychlosti na lokalitě a způsobu připojení**

Posledním měření se týkalo sítě Tor. Ta se ze všech měřených anonymizérů ukázala jako nejpomalejší s poměrně velkým kolísáním a fragmentací přenášených dat, avšak prokázala se jako nejlepší v rámci zabezpečení přenášených dat a jednoduchosti připojení.

Díky Vidalii nebylo nutné upravovat nastavení směrovače, jelikož Tor si bez problému poradí se směrováním a NATem. Zároveň server i klient jsou po celou dobu komunikace chráněni proti zjištění jejich identity.

# 4. ZÁVĚR

Tato práce se zabývala anonymním pohybem uživatele v rámci Internetu. Anonymita je v dnešní době dosti podceňována a spojována s podezřelými aktivitami. Avšak právě pomocí anonymity mohou lidé, kteří mohou být jakýmkoliv způsobem perzekuování, volně komunikovat s vnějším světem a využívat tak svá základní lidská práva.

Hlavní přínos této práce spočívá ve zjištění možných druhů připojení k Internetu tak, aby byla v co největší míře dosažena anonymita a nepostižitelnost uživatele. Práce se věnovala vlastnostem jednotlivých druhů připojení a nalezení jejich slabin.

Dále se podrobně věnovala projektu Tor, který v dnešní době poskytuje nejlepší a nejkomplexnější zabezpečení v rámci anonymního přístupu k Internetu nebo vnitřní síti Tor. Naopak přístup skrze proxy server se ukázal jako dosti nebezpečný, nestabilní a nevhodný k přenosu citlivých dat. Připojení skrze VPN patřilo k jedněm z nejrychlejších, avšak bez možnosti kontroly, zdali si poskytovatel neponechává citlivé informace

Následně byla v práci popsána a demonstrována funkčnost Skryté služby, pomoci nichž lze provozovat server uvnitř sítě Tor tak, aniž by bylo možné determinovat jeho lokaci a tím jej vyřadit z provozu. Byla implementována řešení, která zamezují možné diskreditaci Skryté služby a zároveň umožňuje přístup pouze ověřeným uživatelům.

Další část práce se zabývala implementací anonymní autentizace a zabezpečení přistupujícího klienta tak, aby byla zaručena jeho anonymita a bylo možné umožnit nebo zamezit přístupu na určité části serveru. K tomu bylo využito obousměrné SSL autentifikace, která umožňuje identifikovat obě strany komunikace. Jelikož návrh využívá OpenSSL bylo nutné implementovat nejnovější knihovny, které zamezuje možnému zjištění privátních klíčů a jiných dat díky chybě heartbleed.

Poslední část této práce se zabývala vlivem rychlostí a stability na přenos uživatelských dat, v rámci jednotlivých anononymizátorů. Z nich je patrné, že rychlost spojení je velmi citlivá na přenosovou cestu a množství komunikačních uzlů na její cestě. Zároveň je zřejmé, že s rostoucí mírou zabezpečení klesá podstatně přenosová rychlost.

Díky výsledkům této práce je umožněn zabezpečený přístup uživatele skrze nezabezpečenou síť Internet a umožněna zabezpečená a anonymní autentizace vůči serveru. Právě díky tomuto návrhu lze umožnit přístup a šíření svobodných informacím všem osobám, které by bylo možné identifikovat a následně perzekuovat nebo blokovat při pokusu o jejich získání a šíření.

# LITERATURA


[1] Apache Lounge: Webmasters [online]. 2014 [cit. 2014-04-26]. Dostupné z: http://www.apachelounge.com/

[2] BIRYUKOV, Alex, Ivan PUSTOGAROV a Ralf-Philipp WEINMANN. Trawling for Tor Hidden Services: Detection, Measurement, Deanonymization [online]. 2013 [cit. 2014-04-21]. DOI: 10.1109/SP.2013.15. Dostupné z: http://www.ieee-security.org/TC/SP2013/papers/4977a080.pdf

[3] BridgeDB: The Tor Project. © THE TOR PROJECT. [online]. 2014 [cit. 2014-04-26]. Dostupné z:https://bridges.torproject.org/bridges

[4] CALETKA, Ondřej. Heartbleed bug: vážná zranitelnost v OpenSSL. In: ROOT.CZ [online]. 2014-04-11 [cit. 2014-04-21]. Dostupné z: http://www.root.cz/clanky/heartbleed-bug-vazna-zranitelnost-v-openssl/

[5] DINGLEDINE, Roger, Nick MATHEWSON a Paul SYVERSON. Tor: The Second-Generation Onion Router. [online]. 18.5.2004 [cit. 2013-12-24]. Dostupné z: https://svn.torproject.org/svn/projects/design-paper/tor-design.pdf

[6] Gpg4win: a secure solution... [online]. 2014 [cit. 2014-04-26]. Dostupné z: http://www.gpg4win.org/

[7] HASSAN, Rubayat. Setting up Apache HTTPS/SSL on Windows [online]. 2.4.2010 [cit. 2013-12-24]. Dostupné z: http://rubayathasan.com/tutorial/apache-ssl-on-windows/

[8] HOŘEJŠ, J. Zabezpečení přenosu dat mezi klienty a serverem přes hessian protokol. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2012. 49 s. Vedoucí bakalářské práce Ing. Radim Burget, Ph.D..

[9] CHENG, Elvin. An Introduction to Mutual SSL Authentication. In: Code Project: For those who code [online]. 2012.02.08 [cit. 2014-04-21]. Dostupné z: http://www.codeproject.com/Articles/326574/An-Introduction-to-Mutual-SSL-Authentication

[10] IOERROR. The DigiNotar Debacle, and what you should do about it. In: [online]. 31.8.2011 [cit. 2013-12-24]. Dostupné z: https://blog.torproject.org/blog/diginotar-debacle-and-what-you-should-do-about-it

[11] JANSEN, Rob, Florian TSCHORSCH, Aaron JOHNSON a Bjorn SCHEUERMANN. The Sniper Attack: Anonymously Deanonymizing and Disabling the Tor Network. In: [online]. San Diego, CA, USA, 2014-02-23 [cit. 2014-04-21]. ISBN 1-891562-35-5. Dostupné z: http://www.robgjansen.com/publications/sniper-ndss2014.pdf

[12] JANSEN, Rob. New Tor Denial of Service Attacks and Defenses. In: Blog Tor Project [online]. 2014.01.14 [cit. 2014-04-21]. Dostupné z: https://blog.torproject.org/blog/42

[13] KAMM, Tobias, Thomas LAUTERBACH, Karsten LOESING, Ferdinand RIEGER a Christoph WEINGARTEN. Hidden Service Authentication. In: [online]. 2007-09-10 [cit. 2014-04-21]. Dostupné z: https://gitweb.torproject.org/torspec.git/blob/HEAD:/proposals/121-hidden-service-authentication.txt

[14] LINDELL, Andrew Y. Anonymouse Authentication. In: [online]. Israel: Dept. of Computer Science, Bar-Ilan University, [2007] [cit. 2014-04-21]. Dostupné z:

https://www.blackhat.com/presentations/bh-usa-07/Lindell/Whitepaper/bh-usa-07-lindell-WP.pdf

[15] MACHATA, Tomáš. Tvorba pokročilého simulačního modelu technologie DiffServ v prostředí Opnet Modeler. Brno, 2008/2009. Dostupné z: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=18104 Bakalářská práce. VUT Fakulta elektrotechniky a komunikačních technologií. Vedoucí práce Ing. Jiří Hošek, Ph.D.

[16] O'CEARBHAILL, Donncha. Trawling Tor Hidden Service: Mapping the DHT. Donncha O'Cearbhaill [online]. 2013.03.15, 2013.08.15 [cit. 2014-04-21]. Dostupné z: http://donncha.is/2013/05/trawling-tor-hidden-services/

[17] ØVERLIER, Lasse a Paul SYVERSON. Locating Hidden Servers. In: [online]. 2006 [cit. 2014-04-21]. IEEE Symposium on Security and Privacy. Dostupné z: http://www.onion-router.net/Publications/locating-hidden-servers.pdf

[18] Public Proxy Servers. [online]. © 2002-2014 [cit. 2014-04-26]. Dostupné z:http://www.publicproxyservers.com/

[19] SCHNEIER, Bruce. Schneier on Security: How the NSA Attacks Tor/Firefox Users With QUANTUM and FOXACID. [online]. 7.8.2013 [cit. 2013-12-24]. Dostupné z: https://www.schneier.com/blog/archives/2013/10/how_the_nsa_att.html

[20] SYVERSON, Paul. Onion Routing: Brief Selected History. In: [online]. 2005 [cit. 2013-12-24]. Dostupné z: http://www.onion-router.net/History.html

[21] THE APACHE SOFTWARE FOUNDATION. Apache: HTTP Server Project [online]. © 1997-2014 [cit. 2014-04-26]. Dostupné z: http://httpd.apache.org/

[22] The Heartbleed Bug. CODENOMICON. [online]. 2014-04-15 [cit. 2014-04-21]. Dostupné z: http://heartbleed.com/

[23] The Perl for MS Windows, free of charge! [online]. 2014 [cit. 2014-04-26]. Dostupné z: http://strawberryperl.com/

[24] Tor Project. [online]. [cit. 2013-12-24]. Dostupné z: https://www.torproject.org

[25] TOR PROJECT. Tor Metrics Portal: Network. [online]. [2014] [cit. 2014-04-21]. Dostupné z: https://metrics.torproject.org/network.html

[26] Tor Stinks. In: SNOWDEN, Edward. [online]. červen 2012 [cit. 2013-12-24]. Dostupné z: http://www.theguardian.com/world/interactive/2013/oct/04/tor-stinks-nsa-presentation-document

[27] Tor: Mirrors. TOR PROJECT. [online]. 2014 [cit. 2014-04-26]. Dostupné z: https://www.torproject.org/getinvolved/mirrors.html.en

[28] WINTER, Philipp a Stefan LINDSKOG. Spoiled Onions: Exposing Malicious Tor Exit Relays. In: [online]. Karlstad University, 20.1.2014 [cit. 2014-04-21]. Dostupné z: http://www.cs.kau.se/philwint/spoiled_onions/techreport.pdf

# SEZNAM ZKRATEK

AES Advanced Encryption Standard - Symetrická bloková šifra

CA Certification Authority – Certifikační autorita

CAPTCHA Completely Automated Public Turing test to tell Computers and Humans Apart - Turingův test

CGI Common Gateway Interface – Sktrip, který umožňuje spouštět programy na webovém serveru

CN Common Name – Jméno uživatele nebo adresa stránky nejčastěji FQDN

CPP Client Puzzle Protocol- Protokol pro oddělení počítače od osoby

CSR Certificate Signing Request – Požadavek na podepsání certifikátu

DARPA Defense Advanced Research Projects Agency - Agentura pro výzkum pokročilých obranných projektů

DDoS Distributed Denial of Service - DoS útok uskutečněný velkým počtem počítačů

DNS Domain Name Server - Server umožňující překlad adres

DoS Denial of service - Způsob útoku pomocí odmítnutí služby

DS Directory Server - Rejstříkový server

EFF Electronic Frontier Foundation - Mezinárodní nezisková organizace pro digitální práva

FQDN Fully Qualified Domain Name – Přesná adresa počítače v doménové struktuře

GNU General Public License - Všeobecná veřejná licence

HS Hidden Service - Skrytá služby

HSDir Hidden Service Directory – Databázový adresář Skryté služby

HTTP Hypertext Transfer Protocol - Protokol pro výměnu hypertextových dokumentů

HTTPS Hypertext Transfer Protocol Secure - Zabezpečená nadstavba protokolu HTTP

ICMP Internet Control Message Protocol - Protokol pro odesílání chybových zpráv

ID Identification Data – Identifikační data

IP Internet Protocol - Protokol síťové vrstvy

IP Introduction Point - Bod seznámení v síti Tor neboli RP

IPsec IP security - Bezpečnostní rozšíření IP protokolu

IPv4 Internet Protocol version 4 - Datově orientovaný protokol

IPv6 Internet Protocol version 6 - Datově orientovaný protokol nahrazující IPv4

ISP Internet Service Provider - Poskytovatel Internetu

MiM Man in the middle – Útok člověka uprostřed

MSI Microsoft Windows Installer – Instalační služba pro systém Microsoftu

NAT Network address translation - Překlad síťových adres

NAT-T NAT-Traversal - Verze NATu pro IPsec komunikaci

NSA National Security Agency - Národní bezpečnostní agentura

ONR Office of Naval Research - Úřad spadající pod Námořnictvo Spojených států amerických

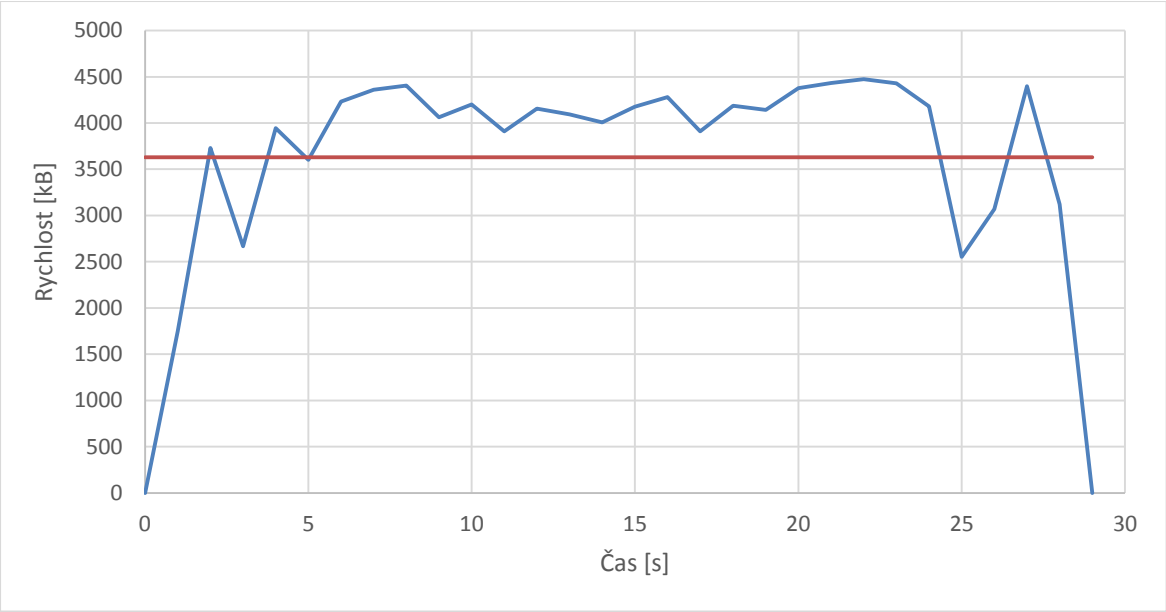
OP Onion Proxy - Součást klienta sítě Tor starající se o spojení

| | |
|---|---|
| OR | Onion Router - Cibulový směrovač v síti Tor |
| OSPF | Open Shortest Path First - Druh směrovacího protokolu |
| P2P | Peer-to-peer - Komunikace typu klient-klient |
| PKCS | Public Key Cryptographic Standards – Standard pro práci s kryptografickými klíči |
| RP | Rendezvous Point - Bod seznámení v síti Tor |
| RSA | Rivest, Shamir, Adleman - Šifra s veřejným klíčem |
| SSH | Secure Shell – Zabezpečený komunikační protokol |
| SSL | Secure Sockets Layer - Vrstva zabezpečení mezi transportní a aplikační vrstvou |
| TCP | Transmission Control Protocol - Protokol transportní vrstvy |
| TDES | Triple Data Encryption Standard – Trojnásobná symetrická bloková šifra DES |
| TLS | Transport Layer Security - Kryptografický protokol pro zabezpečení komunikace |
| TOR | The Onion Router - Cibulový směrovač, název pro síť poskytující anonymitu |
| UDP | User Datagram Protocol - Protokol transportní vrstvy |
| USA | United States of America – Spojené státy americké |
| VPN | Virtual Private Network - Virtuální privátní síť |

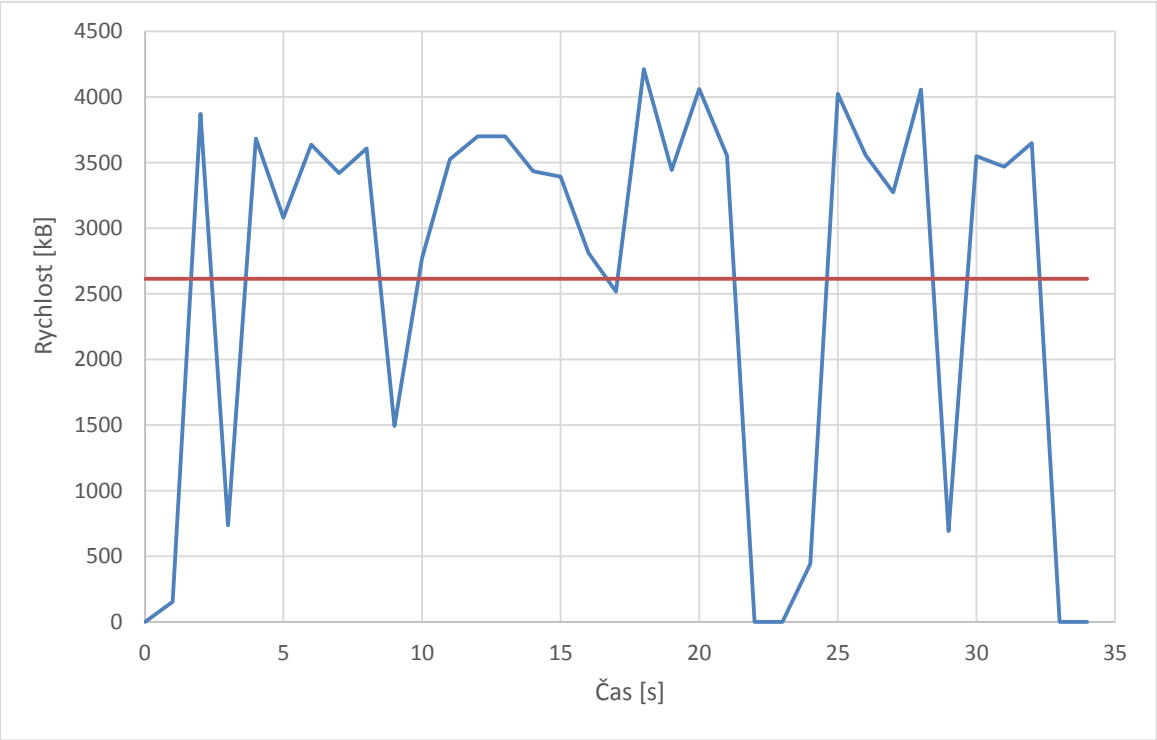
# SEZNAM PŘÍLOH

Obr. A.1. Graf přenosové a průměrné rychlosti v čase pro přímé spojení – 100MB..... 79
Obr. A.2. Graf přenosové a průměrné rychlosti v čase pro VPN – Brno – 100MB ..... 79
Obr. A.3. Graf přenosové a průměrné rychlosti v čase pro VPN – Cork – 100MB..... 80
Obr. A.4. Graf přenosové a průměrné rychlosti v čase pro VPN – Austin – 100MB..... 80
Obr. A.5. Graf přenosové a průměrné rychlosti v čase pro spojení skrze proxy - 100MB ..... 81
Obr. A.6. Graf přenosové a průměrné rychlosti v čase pro síť Tor – 100MB..... 81

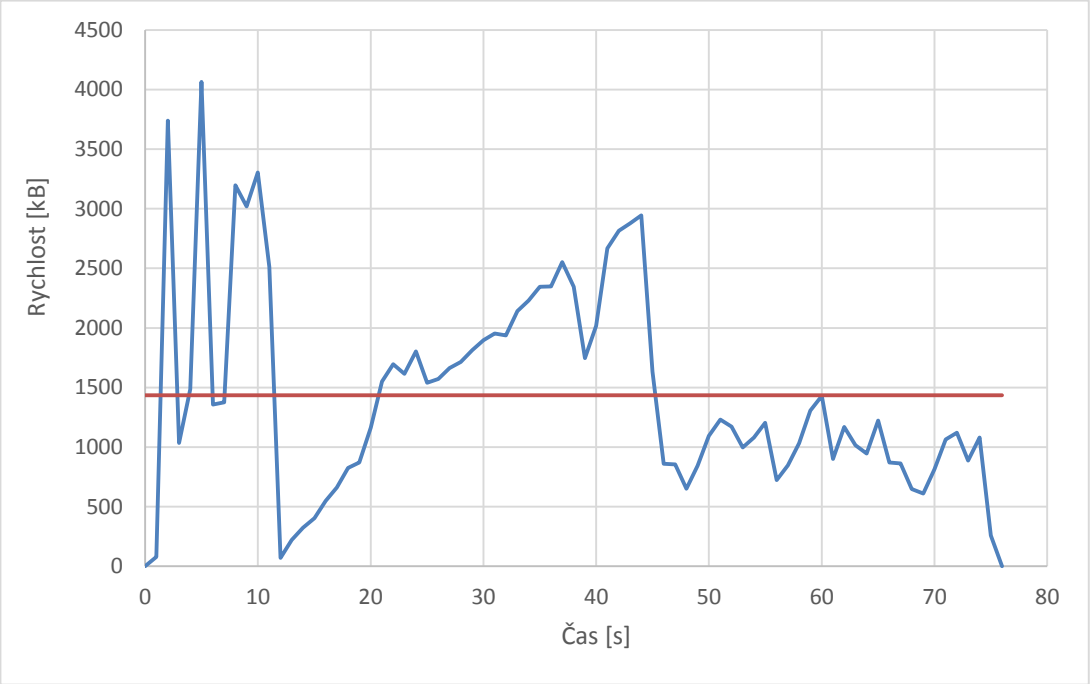
# A PŘÍLOHY



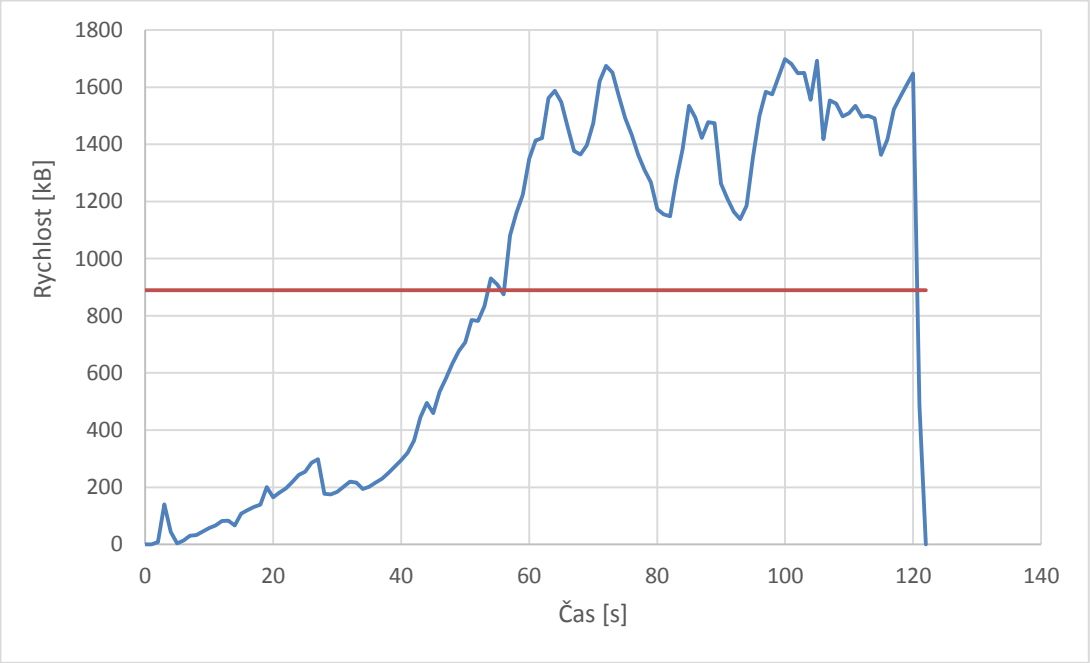

**Obr. A.1. Graf přenosové a průměrné rychlosti v čase pro přímé spojení – 100MB**



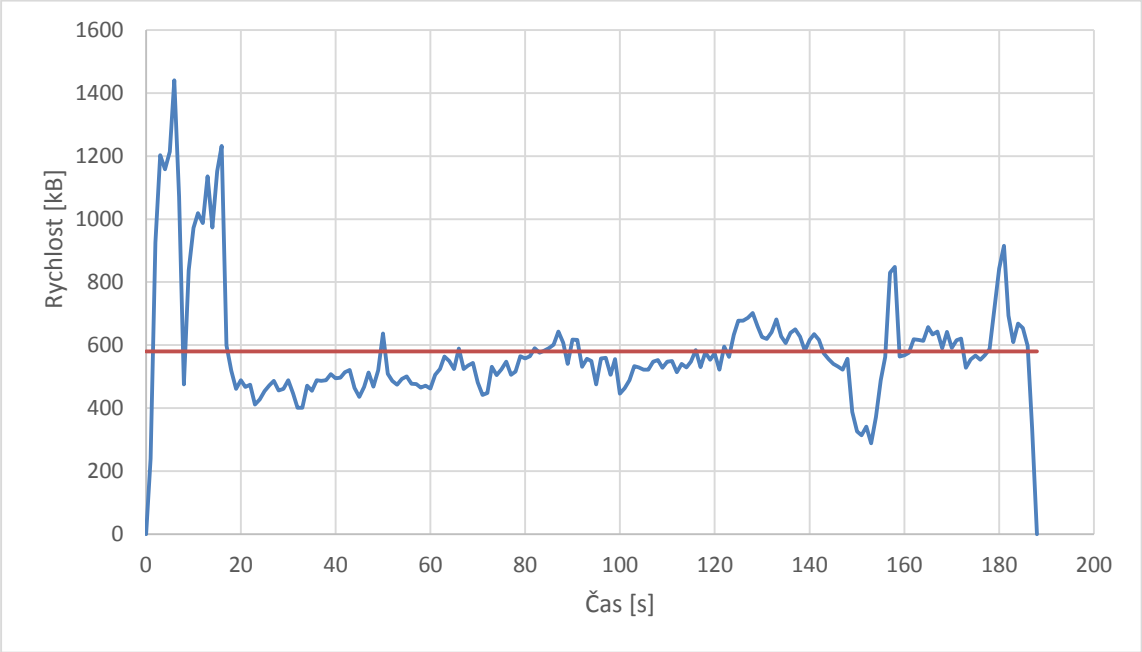

**Obr. A.2. Graf přenosové a průměrné rychlosti v čase pro VPN – Brno – 100MB**



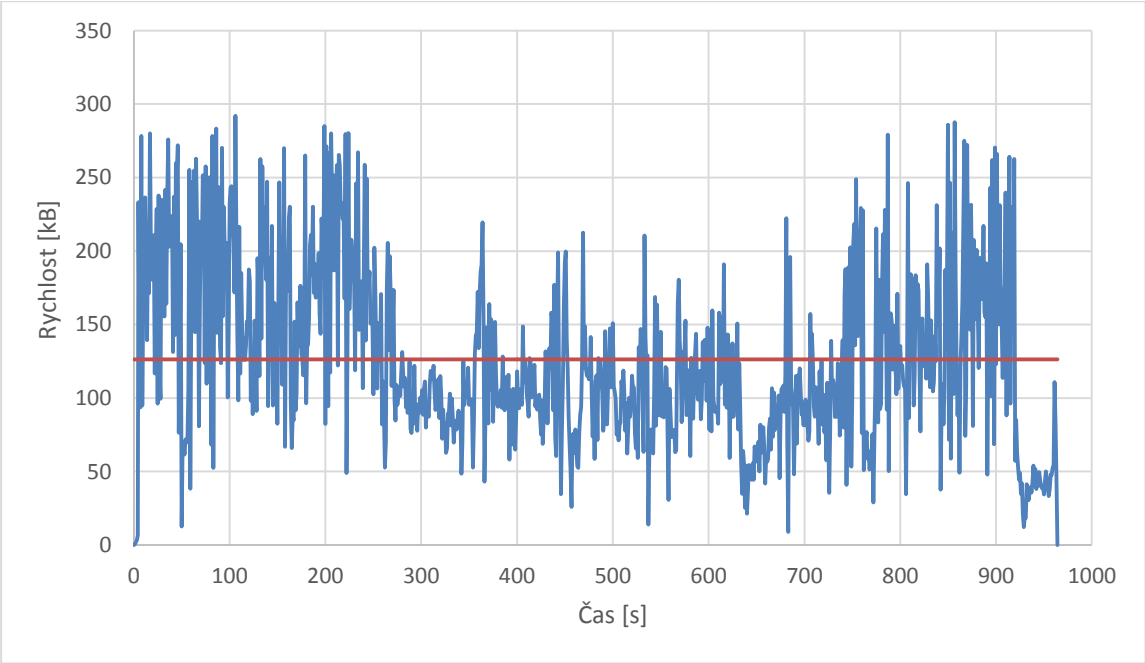

**Obr. A.3. Graf přenosové a průměrné rychlosti v čase pro VPN – Cork – 100MB**




**Obr. A.4. Graf přenosové a průměrné rychlosti v čase pro VPN – Austin – 100MB**




**Obr. A.5. Graf přenosové a průměrné rychlosti v čase pro spojení skrze proxy – 100MB**




**Obr. A.6. Graf přenosové a průměrné rychlosti v čase pro síť Tor – 100MB**

# OBSAH PŘILOŽENÉHO DVD

\\ReadMe.pdf

\\Diplomová práce\

\\Software\

\\Windows Server 2012\