

Off-The-Record protocol a jeho implementace na instant messenger

Off-The-Record Protocol and Its Application in Instant Messenger

Jakub Skřivánek



ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Jakub SKŘIVÁNEK**
Osobní číslo: **A09067**
Studijní program: **B3902 Inženýrská informatika**
Studijní obor: **Informační a řídicí technologie**
Forma studia: **kombinovaná**

Téma práce: **Off-The-Record protocol a jeho implementace na instant messenger**

Zásady pro vypracování:

1. Vypracujte literární rešerši na dané téma.
2. Popište principy šifrování pomocí OTR.
3. Zjistěte současný stav užívání OTR.
4. Popište možnosti a způsoby tzv. instant messaging používaném na Internetu.
5. Aplikujte OTR knihovnu na existující instant messenger.

Rozsah bakalářské práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. VONDRUŠKA, Pavel. Kryptologie, šifrování a tajná písma. Praha: Albatros, 2006. ISBN 8000018888.
2. MOLDOVYAN, Nick a Alex MOLDOVYAN. Innovative cryptography. 2nd ed. Boston: Charles River Media, 2007, xiii, 386 p. ISBN 978-158-4504-672.
3. DENIS, Tom St. a Simon JOHNSON. Cryptography for developers. Rockland, Mass.: Syngress, 2007, xxii, 423 s. ISBN 978-159-7491-044.
4. KATZ, Jonathan a Yehuda LINDELL. Introduction to modern cryptography. Boca Raton: Chapman & Hall/CRC, 2008, xviii, 534 s. ISBN 978-1-58488-551-1.
5. KIAYIAS, Aggelos a Serdar PEHLIVANOGLU. Encryption for digital content. New York: Springer, 2010, xiii, 209 s. ISBN 978-1-4419-0044-9.
6. GOLDBERD, Ian et al. Off-the-Record Messaging [online]. Icit. 2013-05-021. Dostupné z: <http://www.cypherpunks.ca/otr/>

Vedoucí bakalářské práce:

Ing. Roman Šenkeřík, Ph.D.

Ústav informatiky a umělé inteligence

Datum zadání bakalářské práce:

24. února 2013

Termín odevzdání bakalářské práce:

14. června 2013

Ve Zlíně dne 24. února 2013

prof. Ing. Vladimír Vašek, CSc.

děkan



prof. Ing. Vladimír Vašek, CSc.

ředitel ústavu

ABSTRAKT

Tato práce se zabývá šifrováním komunikace, za pomoci Off-the-record protokolu a využití v „instant messaging“ komunikace. V práci je rozpracován teoretický popis protokolu a možností „instant messaging“ komunikace. Práce dále obsahuje popis přípravy prostředí a vlastní implementaci protokolu do již existujícího webového klienta pro instant messaging.

Klíčová slova: Off-the-record, OTR, IM, „instant messaging“, šifrování

ABSTRACT

The bachelor's thesis deals with encryption of communication with Off-the-record protocol and its usage within „instant messaging“ conversation. The thesis presents theoretical description of the protocol and usage of „instant messaging“ conversation itself. Furthermore the thesis contains preparation environment description and the protocol implementation to already existing web client for „instant messaging“.

Keywords: Off-the-record, OTR, IM, „instant messaging“, cryptography

„Assuming of course we're not dealing with five-dimensional objects in a basic Euclidean geometric universe and given the essential premise that all geo-mathematics is based on the hideously limiting notion that one plus one equals two, and not as {Astemeyer} correctly postulates that one and two are in fact the same thing observed from different precepts, the theoretical shape described by {Siddus} must therefore be a poly-dri-doc-deca-wee-hedron-a-hexa-sexa-hedro-adicon-a-di-bi-dolly-he-deca-dodron.

Everything else is poppycock. Isn't that so? “

A. J. Rimmer

Prohlašuji, že

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně

.....
podpis diplomanta

OBSAH

ÚVOD.....	9
I TEORETICKÁ ČÁST.....	10
1 HISTORIE ŠIFROVÁNÍ.....	11
1.1 STEGANOGRAFIE	11
1.1.1 Fyzická steganografie.....	11
1.1.2 Digitální steganografie	12
1.2 KRYPTOGRAFIE	13
1.2.1 Klasická kryptografie	13
1.2.2 Moderní kryptografie	15
1.2.2.1 DES	15
1.2.2.2 Triple DES, 3DES.....	16
1.2.2.3 AES	16
1.2.2.4 RSA.....	16
1.2.2.5 PGP	17
1.2.3 Kvantové počítače a jejich vliv na kryptografii	17
2 PROSTŘEDKY OTR.....	18
2.1 AES	18
2.2 DIFFIE-HELLMAN PROTOKOL	18
2.3 MESSAGE AUTHENTICATION CODE	19
2.4 SHA.....	19
2.5 SOCIALIST MILLIONAIRES‘ PROTOKOL	19
2.6 MAN IN THE MIDDLE ATTACK.....	20
3 POPIS PRINCIPU OTR.....	21
3.1 OTR PROTOKOL V. 1	21
3.1.1 Šifrování.....	21
3.1.2 Zapomínání klíčů	22
3.1.3 Autentizace.....	23
3.1.4 Odhalení MAC klíčů	23
3.1.5 Man in the Middle Attack na OTR protokol v.1	24
3.2 OTR PROTOKOL V. 2	25
3.2.1 Šifrování.....	25
3.2.2 Upravený Socialist Millionaires‘ protocol	27
3.2.2.1 Výběr generátoru	27
3.2.2.2 Skrytí x a y	27
3.2.2.3 Odhalení zda $x=y$	27
4 SOUČASNÝ STAV UŽÍVÁNÍ OTR	29
4.1 DISTRIBUCE OPERAČNÍCH SYSTÉMŮ OBSAHUJÍCÍ SOFTWARE S UŽITÍ OTR	29
4.2 IM KLIENTI S PODPOROU OTR PROTOKOLU „OUT OF THE BOX“	29
4.3 IM KLIENTI S PODPOROU „THIRD-PARTY“ PLUGINŮ	30
4.4 KNIHOVNY S PODPOROU OTR.....	30
5 MOŽNOSTI A ZPŮSOBY TZV. “INSTANT MESSAGING“.....	31

5.1	“INSTANT MESSAGING“ POUŽÍVANÝ NA INTERNETU	31
5.2	PROTOKOLY PRO “INSTANT MESSAGING“	31
5.2.1	Jabber (XMPP)	31
5.2.2	ICQ (I Seek You)	32
5.2.3	GAGU	32
5.2.4	MSNP	32
5.2.5	GoogleTalk, GTalk (XMPP)	33
5.2.6	Skype	33
5.3	KLIENTI A SLUŽBY PRO “INSTANT MESSAGING“	33
5.3.1	Pidgin	33
5.3.2	Skype	34
5.3.3	Windows live messenger	35
5.3.4	QIP	35
5.3.5	ICQ	36
5.3.6	Facebook	36
5.3.7	Miranda	37
5.3.8	Imo.im	38
5.3.9	Gtalk	38
II	PRAKTICKÁ ČÁST	39
6	PROSŘEDKY A PRVKY K IMPLEMENTACI	42
6.1	IM KOMUNIKÁTOR CANDY	42
6.2	OTR KNIHOVNA	45
6.3	APACHE HTTP SERVER PROJECT	46
6.4	OPENFIRE	46
7	IMPLEMENTACE OTR KNIHOVNY NA IM CANDY A OVĚŘENÍ FUNKČNOSTI	48
7.1	IMPLEMENTACE OTR KNIHOVNY	48
7.2	TEST CANDY IM S OTR	48
7.3	HOTOVÉ ŘEŠENÍ OTR PLUG-IN PRO PIDGIN	52
	ZÁVĚR	56
	ZÁVĚR V ANGLIČTINĚ	57
	SEZNAM POUŽITÉ LITERATURY	58
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	62
	SEZNAM OBRÁZKŮ	63
	SEZNAM PŘÍLOH	64
	PŘÍLOHA P I: CD	65

ÚVOD

Komunikace je součástí našeho všednodenního života. Stejně jako potřeba zajistit, aby adresovanou komunikaci měl možnost přečíst pouze adresát. Skrýt takovou komunikaci, či znemožnit její přečtení se snažíme již z dob historie. V dnešní době velká část naší komunikace přesunula na internet, kde její rychlou formu zpráv, zasílaných v reálném čase, označujeme jako tzv. „instant messaging“ a tak hledáme nové prvky, zajišťující její nevyzrazení a udržení soukromí. K tomu, mimo jiné, slouží Off-the-record (OTR) protokol, který přidává výhody jako ověření účastníků komunikace, neprokazatelnosti autorství odeslané zprávy při zachycení třetí stranou, či zaručení nečitelnosti již odeslaných zpráv, při prolomení šifrovacího klíče útočníkem.

První sekce této práce se zabývá stručnou historií skrývání a šifrování komunikace a vývojem těchto postupů do moderní podoby. Součástí této sekce práce je popis prostředků OTR protokolu k jeho šifrování, dodržení jeho vlastností, vysvětluje princip protokolu, jak těchto vlastností dosahuje a aktuální využití. Pátá kapitola popisuje možnosti a způsoby „instant massaging“ komunikace.

Druhá sekce se věnuje prozkoumání, vyzkoušení a implementací OTR protokolu na existující instant messengery.

I. TEORETICKÁ ČÁST

1 HISTORIE ŠIFROVÁNÍ

Tato kapitola se bude věnovat a stručně vysvětlí historii šifrování. Vysvětlí pojmy Steganografie[17] - tedy ukrývání informace i Kryptografie[10] - změny zprávy do alespoň na první pohled nečitelné podoby.

1.1 Steganografie

Úkolem steganografie je skrýt samotnou existenci zprávy, např. do jiné zprávy. Obsah zprávy a zpráva samotná může být zapsána ve srozumitelné podobě, útočník ovšem nesmí poznat, že je zpráva předávána. Steganografie jako termín vznikl z původních řeckých slov steganos (schovaný) a graphein (psát). V historii byla tato metoda hojně využívána, rozvinula se tedy do různých podob. [37]

1.1.1 Fyzická steganografie

Na to jak fyzicky ukrýt zprávu existuje velké množství i již historických nápadů. Starověcí Řekové zapisovali zprávy na voskem pokryté dřevěné desky. Text byl zapsán přímo na desce a po překrytí voskem vypadala tato tabulka nepoužitě. Dalším známým způsobem bylo vytetování zprávy, nebo obrázku se zprávou na holou hlavu posla, jakmile mu vlasy zpět dorostly, zpráva se stala ukrytou. Příjemci zprávy tedy opět stačilo hlavu posla oholit.

Milánský fyzik, astronom a matematik Giraolamo Cardano, navrhl v šestnáctém století systém založený na skrytí otevřeného textu do těla jiné zprávy, ta se může zdát nezasvěceným nezávadná. Cardanova mřížka, jak byl tento systém pojmenován, se zakládá na obdélníkové nebo čtvercové mřížce, do které se prostřihnou určitá pole. Po zapsání utajované zprávy skrz tyto pole se ostatní písmena doplní tak, aby celek dával běžnou zprávu. Příjemce zprávu dešifruje opětovným přiložením tabulky. [37]

Neviditelné inkousty nabídly běžnou formu neviditelného psaní. Od základních inkoustů neviditelných do zahřátí se způsob tohoto ukývání vyvinul až k inkoustům odhalitelných pouze za pomoci určitých chemikálií. Italský vědec G. Porta, popsal v 16. století způsob, jak napsat skrytou zprávu na bílek uvařeného vajíčka skrz skořápku. Zprávu napsal pomocí roztoku octa a ledku a po oloupání byla zpráva viditelná. Zprávy psané neviditelným inkoustem se používaly ještě na začátku druhé světové války, byl to téměř výhradní způsob steganografické technologie. S neviditelným inkoustem, i na pohled nevinný dopis, mohl obsahovat zašifrovanou zprávu napsanou mezi řádky. Tento způsob skrývání zpráv byl

překonán vynálezem univerzální žárovky pro všechny typy neviditelných inkoustů a to tím, že žárovka měnila strukturu vláken nepopsaného papíru a tedy odhalovala popsané části.

V druhé světové válce také využili další způsob skrytí tajné zprávy a to do dalšího, na první pohled běžného textu. Například tak, že se zpráva skládala z každého třetího písmena každého slova. Skrytá zpráva se dá také skrýt do běžného tištěného textu, např. odlišným odsazením důležitých slov. Němci také v období druhé světové války vynalezli technologii mikroteček. Zpráva až o velikosti běžné stánky papíru nebo fotografie se skrývala v tečce o velikosti interpunkčního znaménka. Tato technologie byla americkým ředitelem výzvědných služeb označena za nepřátelské veledílo. [17]

1.1.2 Digitální steganografie

S rozšířením digitálního obsahu, vzniku autorských digitálních děl získala steganografie nový rozměr. Autorská díla se, s rozvojem internetu a přenosných digitálních zařízení, stala častým terčem neoprávněného kopírování – digitálního pirátství a bylo nutno zajistit, že autorův podpis se stane velmi těžce změnitelnou součástí. Potřeba zajištění autorských práv vedla ke vzniku watermark (vodoznaku).

Steganografie se v poslední době stává opět oblíbenou metodou k přenášení tajné informace, byť v digitální podobě a to po tom, co se různé šifrovací algoritmy stávají méně bezpečné s rostoucím výpočetním výkonem a klesající pořizovací cenou osobních a serverových počítačů, nařízení autoritami vkládat do šifrovacích algoritmů zadní vrátka pro rozšifrování, nebo k nepovolenému používání silných šifer, či k automatickému odsouzení porušení autorských práv a přenosu zakázaných dat při odmítnutí rozšifrování digitálního přenosného záznamového zařízení na hranicích určitých států. [17]

Steganografii v digitálních datech využijeme tak, že tajnou informaci ukryjeme do oblastí šumu, redundantních informací, oblastí s bezvýznamným obsahem nebo oblastí s neobsazenými daty. Využíváme skrytí dat mimo strukturu daných souborů, ze kterých jde vyvodit obsah. Do různých typů souborů ukrýváme, v naprosté většině, binární data. Mohou to tedy být text, obrázek, program, nebo zvuk.

Jaké množství binárních dat a jak dobře půjdou ukrýt, záleží na vlastnostech krycího objektu. Krycí data mohou být použita podle typu více způsoby, tajné informace mohou být ukryté také více způsoby, na odlišných místech a v různých variacích, kdy jejich přesná identifikace pak tvoří tajný klíč. Za zajímavou verzi digitální steganografie může být

považován i princip skrytého a zašifrovaného oddílu na dalším šifrovaném oddílu toho stejného záznamového média. Po zadání hesla se zobrazí pouze nezávadné informace, které dále nenاسvědčují o přítomnosti dalších skrytých dat, takže v podstatě slouží jako kamufláž. Tohoto principu využívá program TrueCrypt.[17]

1.2 Kryptografie

Z historického pohledu je to věda o tom, jak navrhovat a také používat šifrovací systémy a šifrovací algoritmy. Tedy jak učinit zprávu nečitelnou i pokud se dostane co nepřátelských rukou, nebo je při procesu doručování zachycena třetí, nepovolanou stranou. Součástí kryptografie jsou kryptologové, starající se o rozvoj algoritmů ke skrytí obsahu zprávy, dále se starající o algoritmy k identifikaci správného odesílatele a k ověření správnosti obsahu zprávy, ale také je součástí kryptoanalýza a kryptoanalytici, kde je cílem rozvoj metod luštění šifrovaných zpráv a šifrovacích systémů, nebo proniknutí do těchto systémů a rozšifrování původní tajné zprávy.[37]

1.2.1 Klasická kryptografie

V historii se často šifrování provádělo nějakým posunem či náhradou znaků. U substitučních šifer nahrazuje znak otevřeného textu znakem šifrovaným. Příjemce tedy musí použít invertovanou substituci, aby získal původní zprávu. Do klasické kryptografie tedy řadíme zvláště 4 typy substitučních šifer. Monoalfabatická substituční šifra, taky označovaná za jednoduchou substituční šifru, nahrazuje každý znak původní zprávy s využitím předpisu stejného pro každé písmeno znakem příslušným z šifrovaného textu. Jde o proudovou šifru, nahrazuje se písmeno po písmenu.

Je to např. Césarova šifra, šifra používaná s posunem o tři znaky. Jednoduché šifry ovšem lehce podléhají frekvenční analýze, která porovnává frekvenci výskytu znaků v původní i šifrované zprávě. V případě velmi krátké zprávy je účinný útok hrubou silou.

Homofonní substituční šifra se podobá jednoduché substituční šifře, pro ztížení odhalení pomocí frekvenční analýzy využívá náhradu nejvíce frekventovaných znaků v jazyce původní zprávy, znakem jedním z několika přiřazených jako varianty.

Za polygramovou substituční šifru označujeme šifry, které nahrazují několika znaků původní zprávy, několika znaky šifrované zprávy. Tato šifra se s rostoucím počtem nahrazovaných znaků stává značně složitou, její substituční tabulka adekvátně roste. Tabulku je potřeba sestavit o velikosti N^k , kde k je počet znaků nahrazované skupiny.

Polyalfabetická šifra postupně aplikuje monoalfabetické transformace na jednotlivé po sobě řazené znaky původní zprávy. Pokud je posloupnost transformací konečná, dojde po několika transformacích k opakování jednotlivé transformace. Pět odlišných substitučních šifer tedy může vytvářet jednu polyalfabetickou. Pro snazší vytvoření polyalfabetické substituční šifry se používají různé postupy a to například Vignérova šifra. Tuto šifru tvoří konečná posloupnost jednoduchých substitučních šifer. Pro zjednodušení zapsání původní zprávy do této šifry se používá Vignerův čtverec.[36]

Kryptografie, tak jako již několikrát v minulosti, sehrála významnou roli v druhé světové válce. Po vzniku a rozšíření rádiového vysílání a tedy i komunikace přes tuto technologii došlo k potřebě vysílané zprávy zašifrovat. Takováto komunikace měla výhody v rychlosti přenosu zprávy a v možnosti zaslání zprávy i do odříznutých oblastí, ovšem značnou nevýhodu tvořila tato snadnost zachycení nepřitelem. Komunikace se tedy ze začátku šifrovala pomocí složitějších substitučních šifer, k tomu se používaly kódové knihy. S vývojem techniky a s rostoucími požadavky na sílu šifrování docházelo k vývoji mechanických, či elektromechanických šifrovacích strojů. Nejznámější a také vytvářející velice silnou šifru byl elektromechanický stroj enigma. Pro zašifrování využíval soustavu kotoučů a obsluhu, po stisku znaku z původní zprávy, informoval o zašifrovaném znaku na světelném panelu. Tento důmyslný stroj stál za mnoha hmotnými ztrátami spojenců a i po získání manuálu k tomu, jak tento stroj používat a jak na něm zašifrovat zprávu, se dlouho nikomu nedařilo zprávu rozšifrovat, bez znalosti nastavení aktuálního k vysílané zprávě. Toto nastavení prováděla obsluha mechanickými změnami propojení elektronické ovládací desky stroje. Spojenci využili nativního jazyka indiánů kmene Navajo, pro jeho značnou odlišnost od všech běžně používaných jazyků. [12]

Jako jedinou, teoreticky neprolomitelnou šifru, známe šifru Vernamovu. Tato šifra je proudově substituční a využívá heslář na jedno použití. Ten je tvořen sledem náhodných čísel. Heslo se smí použít pouze jednou a musí mít stejnou délku jako zpráva. Toto heslo se musí příjemci doručit tajnou cestou a po použití hned zničit. Po výpadku jednoho znaku zašifrované zprávy, již nelze získat zprávu původní a náhodná čísla pro heslo musí být tvořena posloupností, generovanou v přírodních procesech. Při dodržení těchto pravidel by se útočníci, či jiný náhodný posluchač neměl, i při použití masivní výpočetní síly, dobrat původního znění zprávy.[36]

1.2.2 Moderní kryptografie

Od 50. Let 20. století, došlo k rozvoji počítačů. I přes původní velké rozměry a vysokou pořizovací cenu docházelo k postupnému rozšiřování těchto strojů a zájem odborníků a veřejnosti rostl. K průlomů došlo se vznikem osobních počítačů v 70. letech 20. století a jejich rozšíření v 80. letech, vznikla potřeba používat výpočetně náročnější šifrování a vytvořily se tedy normy, algoritmy pro jejich používání.

Symetrické šifry využívají stejný tajný klíč pro zašifrování i rozšifrování. Při výměně tohoto klíče může dojít k úniku a tedy možného narušení soukromí zpráv. Pro jeho výměnu a ztížení jeho odcizení se využívá asymetrických šifer, vznikne tak tajný kanál na veřejném kanálu. Symetrické šifry mají výhodu v rychlosti, asymetrické zase v tom, že k přenosu klíče se využívá veřejného kanálu, protože šifrovací klíč neodpovídá dešifrovacímu, není jej teda potřeba zabezpečovat. Zabezpečit se musí tedy dešifrovací klíč, k jeho výměně mezi zasilateli nedochází, tato úloha je tedy jednodušší. Asymetrické šifry jsou výpočetně náročné, jdou tedy oproti symetrickým velmi pomalé.[37]

1.2.2.1 DES

Původně vyvinut firmou IBM (ta ho označovala jako algoritmus Lucifer) pro civilní sektor, nikdy nesloužil k šifrování utajených vojenských a vládních dat. Od roku 1977 až do vývoje šifrování AES byl nejrozšířenějším symetrickým šifrovacím algoritmem vůbec.

DES existuje a používá se v několika modifikacích. První z nich je Electronic Code Book (ECB) v něm se otevřená zpráva pouze šifruje a tedy stejná otevřená zpráva má pokaždé stejnou šifrovanou variantu. Doporučuje se použití pouze pro krátké texty, anebo klíče.

Další modifikace je vhodná pro přenos zpráv, je označována Cipher Block Chaining. Dochází k použití první šifrované části jako výstupu a zároveň jako sčítací část otevřeného vstupu. Cipher FeedBack mode modifikace jako první zašifruje náhodnou část a ten pomocí modulo 2 přičte s první částí otevřené zprávy. Poslední typ použití Output FeedBack mode, používá šifrovanou zprávu jako zpětnou vazbu pro otevřenou zprávu.

DES byl původně navržen, aby odolal útoku hrubou silou, kdy jeho klíč byl délky 112 bitů, používaná varianta má klíč délky 56 bitů, z toho 8bitů slouží k zabezpečení samotného klíče paritou. Části zpráv jsou rozděleny na 64 bitové bloky a o šifrování a dešifrování se v přístrojích měly starat samostatné čipy. Algoritmus DES se již podařilo prolomit útoky hrubou silou při použití paralelních počítačů, případně pomocí diferenciální a lineární

kryptoanalýzy. V minulosti byla snaha tento šifrovací algoritmus zesílit, hlavně pomocí použití delšího klíče, to vedlo ke vzniku varianty Triple DES.[36]

1.2.2.2 Triple DES, 3DES

je použit klíč o délce 112 bitů, nebo 168 bitů. V této variantě prvně šifruje prvním klíčem, následně dešifruje pomocí dalšího a v třetím použití opět šifruje pomocí již použitého klíče prvního. Pokud je použit klíč délky 168 bitů, znamená to, že v třetím kroku se opět šifrovalo s odlišným klíčem. Tato mnohem bezpečnější varianta, označovaná EDE, je bohužel třikrát tak pomalá oproti originální variantě algoritmu DES s použitím jenom jednoho šifrovacího klíče délky 56 bitů. [40]

1.2.2.3 AES

Šifrovací algoritmus AES si popíšeme v kapitole zabývající se již OTR protokolem.

1.2.2.4 RSA

V roce 1977 vznikl ještě další šifrovací algoritmus RSA. Byl navržen pro šifrování klíčů, výměnu klíčů a následně pro vytvoření elektronického podpisu a to pány Rivest, Shamir a Adelman. Šifra RSA byla chráněna patentem, od jeho vypršení rozšířené použití využito hlavně pro bezpečnou komunikaci. RSA algoritmus je založen na použití velmi velkých čísel a ty jsou tvořeny součinem dvou velkých prvočísel. Klíč si každý uživatel vytvoří vynásobením dvou prvočísel a výsledné číslo uvede jako jeho veřejný klíč na distribučních kanálech jemu vyhovujících. Pokud některému uživateli chce někdo zaslat zprávu, použije jeho veřejný klíč k zašifrování zprávy vložením tohoto klíče do obecné podoby veřejné jednosměrné funkce. Takto veřejně zašifrovanou, ale konkretizovanou, zprávu zašle zvolenému uživateli, který jako jediný zná původní dvě prvočísla, na jejichž základě vznikl jeho veřejný klíč a tímto zprávu rozšifruje. Pomocí faktorizace (zkoušení, která dvě prvočísla uživatel k vytvoření veřejného klíče použil) se dá na tyto prvočísla dojít a pak zprávu neoprávněně rozšifrovat. Tento proces je ovšem velice výpočetně náročný, s rostoucím výkonem počítačů se pro udržení tohoto náskoku matematiky doporučuje používat násobek dvou prvočísel alespoň o velikosti 10^{308} . [40]

Na počátku 80. let 20. století si šifrování pomocí RSA mohly dovolit pouze velké firmy, či státní organizace disponující velkými počítači, s několika násobně vyšším výkonem, než byl dostupný pro jedince. Možnost šifrovat komunikaci, při použití počítače tedy data, by

měli mít všichni, to si myslel Phill Zimmermann. Vymyslel tedy projekt Pretty Good Privacy.

1.2.2.5 PGP

PGP spojuje výhody symetrického šifrování a asymetrického šifrování do jednoho společného projektu, kdy data jsou šifrována pomocí symetrické šifry a klíč je potom zašifrován pomocí RSA, tedy asymetrické šifry. Spojuje tedy bezpečnost asymetrické šifry s rychlostí symetrické. Klíč pro symetrickou šifru tedy může být bezpečně šířen přes veřejné kanály. PGP je možné používat i pro funkci digitálních podpisů, proces šifrování a dešifrování proběhne opačně. Soukromým klíčem se pak tedy šifruje a veřejným dešifruje, zprávu je možné dešifrovat pomocí veřejného klíče. Tímto postupem se potom dá zaručit ověření autorství. OpenPGP je později zvolený standart pro internetové použití. Autorovi PGP tento výtvar nepřinesl žádné peníze, po tom, co jej z obavy o zakázání zveřejnil na předchůdci [www Usenetu](http://www.usenet.com), byl stíhán americkou vládou pro nelegální šíření zbraní, pod které v té době šifrovací nástroje patřily.[36]

1.2.3 Kvantové počítače a jejich vliv na kryptografii

Všechny šifry teoreticky jednou podlehnou útoku hrubou silou, až na Vernamovu, a to pomocí výpočetního výkonu kvantového počítače. Tato technologie je nám zatím vzdálená. Kryptoanalytici se těší, až se jim pomocí této technologie podaří prolomit každou šifru. Kryptologové zase, až s pomocí kvantové kryptografie dosáhnou dokonalého soukromí, kdy síla šifry bude stát na fyzikálním jevu, pokud se kryptoanalytik pokusí o zachycení zprávy, oba uživatelé již budou varováni, tak zní kvantová teorie. [33]

2 PROSTŘEDKY OTR

V této kapitole budou popsány prostředky, které OTR[9] protokol využívá. Jsou to AES[19], Diffie-Hellman[7], MAC[32], SHA[34], Socialist Millionaires' protokol[27].

2.1 AES

Šifra AES velmi dobře kombinuje výkonnost, bezpečnost, jednoduchost a její flexibilita použití a celková pružnost řešení jsou další výhody, které stály za jejím zvolením jako nový šifrovací standard Organizací NIST. Šifra AES může být označovaná i jako RIJNDAEL a to podle složení písmen z jmen autorů Vincenta Rijamena a Joany Daemenové. Tato šifra díky dobré odolnosti vůči prolomení i včetně časového útoku, by měla vytlačit používání standardu DES.

Symetrická bloková šifra AES pracuje s klíčem i blokem o proměnné velikosti a to i nezávisle na sobě, 128 bitů, 192 bitů či 256 bitů. Používaná varianta využívá velikost bloku, pro vstup i výstup, 128 bitů. [40]

2.2 Diffie-Hellman protokol

Tento protokol, vynalezený v roce 1976, se oproti dřívějším lišil. Náhle nevyžadoval žádný předchozí kontakt mezi odesílatelem zprávy a jejím příjemcem. Odesílatel totiž svou původní zprávu zamkl svým klíčem, využil veřejný algoritmus a zprávu odeslal. Příjemce zprávu přijal, zamkl svým klíčem a poslal zpět odesílateli. Ten odstranil svůj zámek a stále tajnou zprávu odeslal příjemci, který ji odemkl již vlastním klíčem. Tímto tedy došlo k zašifrování zprávy pomocí dvou klíčů. Protokol, vymyšlený pány Diffie a Hellman má i výhody v tom, že lze jednoduše připojit nové členy, informace nutné k zasílání zpráv mohou být umístěny na veřejném místě. [7]

Princip DH protokolu: Předpokládejme, že Alice a Bob se chtějí dohodnout na sdíleném zabezpečeném klíči pomocí DH protokolu na výměnu klíčů. První Alice vygeneruje náhodnou soukromou hodnotu a a Bob vygeneruje náhodnou soukromou hodnotu b . Oba a i b jsou zvoleny z množiny celých čísel. Potom odvodí jejich veřejné hodnoty pomocí parametrů p a g a jejich soukromých hodnot. Alicina veřejná hodnota je $g^a \bmod p$ a Bobova $g^b \bmod p$. Alice i Bob si vymění své veřejné hodnoty. Alice spočítá

$g^{ab} = (g^b)^a \bmod p$ a Bob $g^{ba} = (g^a)^b \bmod p$. Když $g^{ab} = g^{ba} = k$, Alice a Bob provedli výměnu soukromého klíče k . Tento soukromý klíč se využije ke generování krátkodobých šifrovacích klíčů.

2.3 Message Authentication Code

MAC je autentizující štítek, kterým chceme zaručit, že zpráva je originální, neporušená. Zprávu tedy nešifruje, pouze zabezpečuje. MAC má několik typů, často využívaným je HMAC (Hash function-based MAC) používá klíč, nebo klíče ve spojení s hash funkcí k vytvoření checksum, který je přidán ke zprávě.[32] [2] Alice M algoritmem A spočítá hodnoty MAC pro svou zprávu pomocí svojí kopie MAC klíče k a zašle je společně se svojí zprávou. Bob spočítá, pomocí sdíleného klíče, MAC příchozí zprávy a vyhodnotí tak její originalitu, zda není porušená.

2.4 SHA

Secure hash algorithm[34], jednosměrná funkce, kde na vstupu je zpráva (blok proměnné délky) a na výstupu je blok pevné délky 128, 160, 224, 256, 348, nebo 512 bitů, který označujeme hash. SHA standard byl zvolen roku 1993 v The National Institute of Standards and Technology (NIST).

Výhody SHA:

- libovolnou sebemenší změnou v původním souboru dojde ke změně hash hodnoty, tím zkontrolujeme integritu zprávy
- hash hodnota je pro každý dokument jedinečná, teoreticky nemůže tedy nastat situace, že by dva dokumenty měli stejnou hash hodnotu. Tato bezkoliznost v praxi nelze 100% dosáhnout, po nalezení způsobu napadení bezkoliznosti byly označeny hashovací funkce MD4 a MD5 za málo bezpečné a byly nahrazeny funkcemi SHA-1, SHA-2 a SHA-3. Doporučené je používat nejméně SHA-2.

2.5 Socialist Millionaires' protokol

Dva milionáři chtějí porovnat a zjistit, který z nich je bohatší. Nechtějí ale své bohatství svému protivníkovi ukázat. V kryptografii tento protokol používáme k ověření identity partnera pomocí sdíleného tajemství. Takto jej využívá i OTR.

Alice a Bob znají tajné informace o x a y . Chtějí porovnat, zda X je stejné jako Y . SM protokol jim to umožní, aniž by odhalil jiné hodnoty než $(x == y)$. Z pohledu OTR, tajemství obsahuje dlouhodobé autentizující veřejné klíče obou stran, stejně jako informace vložené oběma uživateli. Pokud $x = y$ tak to znamená, že Alice a Bob vložili stejné tajné informace a tak to musí být ty stejné osoby, které stanovili tajemství na začátku. [27]

2.6 Man in the Middle Attack

Neboli útok člověka uprostřed.[22] Je to útok, kdy se útočník nabourá nepozorovaně do konverzace a tváří se jako jedna či druhá strana. Původní strany jsou stále přesvědčeny, že komunikují spolu, útočník, připojený mezi ně, tak může sledovat veškerou konverzaci a při vzájemném zaslání klíčů tuto komunikaci také může úspěšně infikovat.

Alice vyšle zprávou $Z1$ žádost o Bobův klíč k_B , aby mu zaslala tajnou zprávu $Z2$

Alice $Z1, Sign_A \rightarrow MITM Z1, Sign_A \rightarrow Bob$

Alice $\leftarrow k_M MITM k_B \leftarrow Bob$

Alice $g^{k_M}(Z2) \rightarrow MITM g^{k_B}(Z2a) \rightarrow Bob$

Alice je přesvědčena, že komunikuje s Bobem a že mu zaslala tajnou zprávu $Z2$, Bob je přesvědčen, že zpráva $Z2$ a pochází od Alice.

Pro to, aby si byli jisti, že jejich komunikace nebude napadena MITM útokem, mohou počáteční výměnu klíčů zabezpečit dalšími způsoby:

- Vymění si klíče při osobním setkání – jiný bezpečný kanál
- Použijí k výměně technologii, kterou je nemožné napadnout. Např. kvantovou kryptografií
- Ověří svou identitu u nezávislého institutu

3 POPIS PRINCIPU OTR

Tato kapitola se bude zabývat verzemi protokolu OTR, na základě jaké myšlenky vznikly, jaký byl vývoj OTR protokolu a popíše princip OTR protokolu[9].

Nikita Borisov a Ian Goldberg přemýšleli o komunikaci mimo záznam již v roce 2004 a popsali ji v prezentaci z roku 2005 Off-the-Record Communication, or, Why Not to Use PGP, tedy Komunikace mimo záznam a proč nepoužít PGP. Verze protokolu byla označena jako v.1.

Následně došlo na verzi v.2, která řešila možného útoku na prvotní výměnu DH klíče v OTR, pomocí Diffie et al.'s identity misbinding attack [8] pány Di Raimondo, Gennaro a Krawczyk [6]. S druhou verzí nemusí uživatel znát klíče či fingerprinty.

Třetí verze protokolu v.3 přináší extra symetrický klíč v AKE (dovoluje další typy komunikace – hlasový chat, přenos souborů) a vylepšení pro síť IM, které dovolují přihlášení z více klientů naráz. Nyní se OTR klienti pokusí navázat automaticky OTR spojení pro zprávy ze všech aktivních klientů.

3.1 OTR protokol v. 1

3.1.1 Šifrování

Zprávu chceme označit za soukromou, a tedy ji zašifrujeme. Použijeme proudovou šifru AES (viz 2.1). Sdílené tajemství se určí s pomocí volby Diffie-Hellman šifrovacího klíče. Pro zajištění obnovy klíčů, mohou uživatelé zvolit zapomenutí klíčů starých. Uživatelé, Alice a Bob, tedy zruší hodnoty x_A a x_B . Po tomto momentu bude staré zprávy již nemožné rozšifrovat i za pomoci znalosti přenesených hodnot g^{x_A} a g^{x_B} . To označujeme jako perfektní forward secrecy a veškeré staré zprávy se stanou nečitelné.

Díky malé procesorové náročnosti Diffie Hellman algoritmu se mohou klíče měnit s každou zprávou, osobní asistenty, PDA, by měly být schopny provést změnu klíčů alespoň jednou za minutu. Každá zpráva nese informaci DH veřejného klíče g^x který volí klíč pro novou zprávu.

Výměna zpráv tedy může být:

$$A \rightarrow B: g^{x_1}$$

$$B \rightarrow A: g^{y_1}$$

$$A \rightarrow B: g^{x_2}, E(M_1, k_{11})$$

$$B \rightarrow A: g^{y_2}, E(M_2, k_{21})$$

$$A \rightarrow B: g^{x_3}, E(M_3, k_{22})$$

$k_{ij} = H(g^{x_i y_j})$ je výsledek 128 - bitového šifrování H , například SHA-1, na elementu Z_p^* a $E(M, k)$ je uvedeno šifrování AES v counter modu používající klíč k . Každá zpráva je zašifrována pomocí sdíleného tajemství a posledního klíče druhé strany, zaslaného v poslední zprávě a posledním klíčem který byl zaslán druhé straně v poslední odeslané zprávě. Nepoužijeme klíč z jedné zprávy až do následující zprávy. Aby oba účastníci komunikace měli jistotu, který klíč k_{ij} se používá, je potřeba zavést ID hodnotu klíče a také ji přenášet. Není totiž vyžadováno, aby po každé zprávě od Alice následovala zpráva od Boba. Jak je tedy uvedeno v příkladu, klíč v poslední zprávě od Alice je šifrován $H(g^{x_2 y_2})$.

3.1.2 Zapomínání klíčů

Jakmile dojde ke správné výměně klíčů, je nutné ty staré zapomenout, to je podmínka forward secrecy. Protože Alice může zaslat několik zpráv za sebou například 3, Bob může odpovědět s klíčem z prvního kola, aby Alice byla schopná zprávu přečíst (rozšifrovat), musí si stále pamatovat klíč z prvního kola a to až do chvíle, než od Boba získá zprávu zašifrovanou za pomoci této hodnoty. Teoretické ideální zapomenutí požitého klíče ihned po odeslání zprávy tedy nejde použít. Aby nedošlo k tvoření velké posloupnosti klíčů, pokud Bob chvíli na zprávy nereaguje, je Alici povoleno vytvoření nového klíče, až Bob odpoví. Když Alice od Boba přijme zprávu šifrovanou s pomocí klíče g^{x_n} , zapomíná klíč $g^{x_{n-1}}$ a generuje klíč $g^{x_{n+1}}$ pro další zprávu. Tím to se ovšem otvírá prostor oslabení pro možné rozšifrování zpráv od Alice, pokud Bob dlouho neodpovídá. Od Boba se tedy očekává, že by měl alespoň jednou za čas poslat zprávu, kdyby prázdnou, aby došlo k obnově klíčů a riziko oslabení zabezpečení zpráv se snížilo.

3.1.3 Autentizace

V kapitole Prostředky OTR je popsána funkce MAC (viz. 2.3), která autentizaci obstarává.

K vytvoření MAC klíče dojdeme aplikováním jednosměrné hash funkce na dešifrovací klíč. To zaručí možnost změny a obnovení MAC klíče všem, kdo je schopný přečíst zprávu. Útočník pak nemůže nikomu dokázat, že zprávu poslal, mohl to být jak on sám, tak i Alice nebo Bob.

Šifrovací klíč je sám o sobě vznikl z hashe DH sdíleného tajemství. Musí tedy být také autentizován a to pomocí digitálního podpisu prvotní DH výměny.

$$A \rightarrow B: \text{Sign}(g^{x_1}, k_A), K_A$$

$$B \rightarrow A: \text{Sign}(g^{y_1}, k_B), K_B$$

Kde jsou Privátní klíč Alice: k_A a K_A je dlouhodobý veřejný klíč. k_B a K_B patří Bobovi.

Pokud Bob již zná veřejný klíč Alice, bude ujištěn, že g^{x_1} opravdu od Alice přišla a že sdílené tajemství $g^{x_1 y_1}$ bude známo pouze jim. Zpráva ověřená klíčem $H(g^{x_1 y_1})$ pak může být považována jako opravdu zasláná od Alice. Digitální podpisy a MAC zaručují hybridní autentizaci. Sdílené tajemství je pokaždé vytvořeno za běhu, jakmile je potřeba.

Digitální podpisy potřebujeme pouze pro prvotní výměnu klíčů, pro další výměnu se používá MAC, autentizuje nový klíč pomocí starého, známého ověřeného sdíleného tajemství.

Zpráva pak vypadá:

$$g^{x_{i+1}}, E(M_r, k_{ij}),$$

$$\text{MAC}(\{g^{x_{i+1}}, E(M_r, k_{ij})\}H(k_{ij}))$$

Takže pokud prvotní výměna autentizace klíče je známa jako bezpečná, následující budou také.

3.1.4 Odhalení MAC klíčů

Pro další rozměr bezpečnosti Alice zná veškeré zprávy, které poslala Bobovi – ty byly ověřeny jedním MAC klíčem. Alice tento klíč připojí k následující zprávě, když Bob

všechny zprávy už přečetl. V tento moment může kdokoliv, i útočník dodatečně poslat zprávu s použitím tohoto klíče, nikomu ale nemůže být prokázáno autorství.

3.1.5 Man in the Middle Attack na OTR protokol v.1

Útočník (MITM) naruší prvotní výměnu klíčů. MITM spustí komunikaci s Alicí i Bobem. Zprávy od Alice MITM nahradí identickými s vlastním podpisem. Komunikaci od Boba MITM nechá stejnou a nesou originální podpis.

Komunikace pak je:

$$A \rightarrow MITM: g^x, Sign_{S_A}(g^x), v_A$$

$$MITM \rightarrow B: g^x, Sign_{S_{MITM}}(g^x), v_{MITM}$$

$$B \rightarrow MITM: g^y, Sign_{S_B}(g^y), v_B$$

$$MITM \rightarrow A: g^y, Sign_{S_B}(g^y), v_B$$

Alice stále získává zprávy podepsané Bobem a tak si správně myslí, že mluví s ním. Kdežto Bob získává zprávy s podpisem MITM a začne se domnívat, že mluví s MITM, když ve skutečnosti mluví s Alicí.

K zamezení útoku MITM byl implementován SIGMA protokol [21]. Nyní obě strany čekají, až je sestaveno sdílené tajemství g^{xy} a následně zašlou zprávu, kde se identifikují. Protože MITM nezná sdílené tajemství, nemůže v MITM útoku pokračovat. Alice i Bob nyní vzájemně určí svou identitu:

$$A \rightarrow B : g^x$$

$$B \rightarrow A : g^y$$

$$A \rightarrow B: A, Sign_{S_A}(g^y, g^x), MAC_{K_m}(0, A) v_A$$

$$B \rightarrow A: B, Sign_{S_B}(g^x, g^y), MAC_{K_m}(1, B) v_B$$

MACK klíč K_m je hashem g^{xy} a tudíž neznámý MITM. Zahrnutím tohoto MAC, SIGMA je zabráněno útoku identity misbinding.

3.2 OTR protokol v. 2

3.2.1 Šifrování

Vylepšení verze 2 spočívá v nahrazení prvotní výměny klíčů variantou SIGMA. Nyní OTR skrývá i veřejné klíče účastníků komunikace před pasivními útočníky. Veřejné klíče jsou nyní šifrovány pomocí Diffie-Hellman sdíleného tajemství. Protokol stále funguje tak, že nejdříve sestaví neověřený DH kanál a v něm provádí autentizaci. Kanál je použit 64 bitový a vzniknul na základě sdíleného tajemství. 64 bitů je málo, aby zde vzniklo oslabení pro útok hrubou silou. Výsledkem je prověření závazku (commitment), že žádná strana nezaloží g^x na hodnotě druhého účastníka g^y . První krok autentizace výměny klíčů pak je:

Alice:

1. Náhodně vybere 128 bitovou hodnotu čísla r
2. Náhodně vybere 320 bitovou hodnotu čísla x
3. Bobovi zašle $AES(g^x)$, $SHA - 256(g^x)$

Bob:

1. Náhodně vybere 320 bitovou hodnotu čísla y
2. Alici zašle g^y

Alice:

1. Vypočítá hodnotu pro $s = (g^y)^x$
2. Bobovi zašle r

Bob:

1. Rozšifruje g^x s pomocí r
2. Překontroluje, zda souhlasí již přijaté $SHA - 256(g^x)$ s právě přijatým g^x
3. Vypočítá hodnotu pro $s = (g^x)^y$

Smysl použitého r v příkladě nahoře je uspokojit technické restriky – protokoly pro IM komunikaci podporují pouze použití určité maximální délky zprávy. Pro teoreticky ideální případ by Alice jako třetí krok zaslala nejdříve Bobovi $SHA - 256(g^x)$ jako závazek a posléze jako druhý krok by zaslala g^x na otevření závazku. S pomocí r zašifrujeme hodnotu g^x a toto r zašleme první zprávou, ale Alice ho odhalí Bobovi až ve zprávě druhé.

Zašifrovaná verze g^x se nechová jako závazek sama, tuto funkci stále plní hash. Následuje zavedení sdíleného tajemství s . Zs se spočítá s pomocí SHA-256 hash pro Alici i Boba. S přidáním různých prefixů dojde k vytvoření série MAC a AES klíčů. S jejich pomocí se pak zašifrují a ověří integrity informací vyměněných v zbytku průběhu AKE.

Pokud Alicin pár privátních klíčů je (v_A, s_A) a Bobův (v_B, s_B) protokol pokračuje:

Alice:

1. Vypočítá hodnoty MAC klíčů a_1, a_2, b_1, b_2 a AES klíčů a_3, b_3
2. Zvolí $keyid_A$ sériové číslo pro g^x
3. Vypočítá $M_A = MAC_{a_1}(g^x, g^y, v_A, keyid_A)$
4. Vypočítá $X_A = v_A, keyid_A, sign_{s_A}(M_A)$
5. Bobovi zašle $AES_{a_3}(X_A), MAC_{a_2}(AES_{a_3}(X_A))$

Bob:

1. Vypočítá hodnoty MAC klíčů a_1, a_2, b_1, b_2 a AES klíčů a_3, b_3
2. Pomocí a_2 ověří $MAC_{a_2}(AES_{a_3}(X_A))$
3. Pomocí a_3 rozšifruje $AES_{a_3}(X_A)$ a získá $X_A = v_A, keyid_A, sign_{s_A}(M_A)$
4. Vypočítá $M_A = MAC_{a_1}(g^x, g^y, v_A, keyid_A)$
5. Pomocí v_A ověří $sign_{s_A}(M_A)$
6. Zvolí $keyid_B$ sériové číslo pro g^x
7. Vypočítá $M_B = MAC_{b_1}(g^y, g^x, v_B, keyid_B)$
8. Vypočítá $X_B = v_B, keyid_B, sign_{s_B}(M_B)$
9. Alici zašle $AES_{b_3}(X_B), MAC_{b_2}(AES_{b_3}(X_B))$

Alice:

1. Pomocí b_2 ověří $MAC_{b_2}(AES_{b_3}(X_B))$
2. Pomocí b_3 rozšifruje $AES_{b_3}(X_B)$ a získá $X_B = v_B, keyid_B, sign_{s_B}(M_B)$
3. Vypočítá $M_B = MAC_{b_1}(g^y, g^x, v_B, keyid_B)$
4. Pomocí v_B ověří $sign_{s_B}(M_B)$

Protokol je u konce, Alice a Bob si jak vyměnili key ID tak znají sdílené tajemství. Nyní již může proces výměny klíčů probíhat tak, jak je vysvětleno v protokolu verze 1, sekci šifrování (3.1.1.). Alici je také znám Bobův veřejný klíč v_B a je přesvědčena, že on zná odpovídající soukromý klíč s_B . Stejně tak je obeznámen Bob. A protože jsou veškeré hodnoty šifrovány, pasivní útočník by neměl znamenat hrozbu.

3.2.2 Upravený Socialist Millionaires' protocol

Druhá verze OTR protokolu musela být vůči MITM útoku upravena takto:

3.2.2.1 Výběr generátoru

Jsou zavedeny dva přídatné generátory g_2, g_3 , které jsou vytvořeny pomocí DH výměny.

- Alice vybere hodnotu a_2 z množiny \mathbb{Z}_q a Bob si vybere hodnotu b_2 z množiny \mathbb{Z}_q
- Alice a Bob si vymění $g_1^{a_2}$ a $g_1^{b_2}$ a každý vypočítá $g_2 = g_1^{a_2 b_2}$
- Opakují kroky 1 a 2, volí si v nich nové hodnoty a_3 a b_3 , vzájemně si zašlou $g_1^{a_3}$ a $g_1^{b_3}$, aby získali $g_3 = g_1^{a_3 b_3}$.
- Alice a Bob si musí uložit hodnoty a_3 a b_3 , které použily při generování i g_3 , tak aby je mohli využít dále v protokolu.

3.2.2.2 Skrytí x a y

- Alice si vybere hodnotu a z množiny \mathbb{Z}_q a vypočítá $(P_a, Q_a) = (g_3^a, g_1^a, g_2^x)$.
- Bob si obdobně vybere b a vypočítá (P_b, Q_b)
- Dojde k výměně P_a, Q_a, P_b a Q_b

3.2.2.3 Odhalení zda $x=y$

- Alice si pomocí hodnoty a_3 z 3.2.2.1 vypočítá hodnotu $R_a = \left(\frac{Q_a}{Q_b}\right)^{a_3}$
- Bob obdobně spočítá svou hodnotu $R_b = \left(\frac{Q_a}{Q_b}\right)^{b_3}$
- Dojde k výměně R_a a R_b
- Alice i Bob nyní mohou vypočítat $R_{ab} = R_a^{b_3} = R_b^{a_3}$
- Oběma stranám je v tomto momentě známo:

$$\begin{aligned}
 R_{ab} &= \left(\frac{Q_a}{Q_b} \right)^{a_3 b_3} \\
 &= (g_1^{a-b} g_2^{x-y})^{a_3 b_3} \\
 &= g_3^{a-b} g_2^{(x-y) a_3 b_3} \\
 &= \left(\frac{P_a}{P_b} \right) (g_2^{a_2 b_3})^{(x-y)}
 \end{aligned}$$

Pokud se zkontroluje zda $R_{ab} = \left(\frac{P_a}{P_b} \right)$ a dojde ke zjištění, že rovnost platí, musí se pak rovnat i hodnoty x a y .

Přitom nedošlo k odhalení hodnoty $g_2^{a_3 b_3}$ ani jedné z komunikujících stran a také to, že tato hodnota je náhodný generátor G . Z toho plyne, že pokud $x \neq y$, $\left(R_{ab}, \frac{P_a}{P_b} \right)$ potom bude náhodný element generátoru G a pokud $x = y$, pak to bude 1. Protože náhodnost $\left(R_{ab}, \frac{P_a}{P_b} \right)$ není závislá na entropii x a y , pracuje tento protokol velmi dobře, i když je tato entropie velmi malá. Pak to tedy mohou být i jen obyčejná běžná slova.

4 SOUČASNÝ STAV UŽÍVÁNÍ OTR

K prvnímu užití OTR protokolu došlo krátce po jeho uvedení a to přímo jeho stvořiteli.

Plánovali jej využít pro komunikaci známou jako Instant messaging. [11]

Protokol je projektován na použití u klienta, ne na straně serveru, tedy není závislý na IM protokolu. Lze jej využít pro komunikaci přes protokoly Jabber, ICQ, AOL, AIM a jiné.

OTR protokol tedy implementovali jako rozšíření do komunikátoru pidgin. Výhodou použití OTR protokolu na samotného klienta a ne protokol komunikace je, že pokud druhá strana nemá OTR protokol implementován, komunikace může stále probíhat po nezabezpečeném kanálu. Šifrování také probíhá na straně klienta, přes použitý komunikační protokol tedy chodí sice šifrovaná komunikace, ale typu běžných zpráv.

4.1 Distribuce operačních systémů obsahující software s užití OTR

Většina OS, ve kterých je uložen software využívající OTR protokol přímo v instalačním balíčku, jsou Linuxové distribuce. A to:

- Gentoo
- Debian (stable, testing, unstable)
- Ubuntu (Dapper, Feisty, Gutsy, Hardy, Intrepid)
- T2
- Fedora, Red Hat Enterprise Linux, CentOS

Ještě v jednom typu OS je OTR protokol přímo v instalačním balíku. A to v BSD, konkrétně těchto distribucích:

- FreeBSD
- NetBSD
- OpenBSD current
- OpenBSD 4.4

4.2 IM klienti s podporou OTR protokolu „out of the box“

IM komunikátory, které mají OTR protokol přímo zaimplementován a jsou:

- MAC OS IM komunikátor Adium

- Linuxový Kopete, CenterIM
- IM+ pro android a Gibberbot pro android
- Multiplatformní climm, MCABBER a jitsi

4.3 IM klienti s podporou „Third-party“ pluginů

Někteří IM klienti se dočkaly rozšíření o možnou OTR komunikaci pomocí externích (Third-party) developerů. Rozšíření je pak možné za pomoci pluginů, které se do daného klienta nainstalují. Jsou to:

- Multiplatformní Pidgin
- Miranda a Trillian pro Windows
- Linuxový WeeChat

4.4 Knihovny s podporou OTR

Dostupné jsou již i samotné knihovny:

- Libotr
- Python bindings to libotr
- Nativní knihovna pro Python
- Nativní knihovna pro Scheme
- Nativní knihovna pro Java (SE a ME)

5 MOŽNOSTI A ZPŮSOBY TZV. \“INSTANT MESSAGING\“

Kapitola představí jaký typ komunikace se za pojmem „instant messaging“ skrývá, jaké protokoly jsou pro tento typ komunikace potřeba a přes které programy, či služby je možno komunikaci uskutečnit.

5.1 \“instant Messaging\“ používaný na internetu

S rozšířením internetového připojení a hlavně s příchodem nevytáčeného internetového připojení, instant messaging (IM) přivedl revoluci do zavedších zvyklostí internetové komunikace. Vznikl v roce 1988 (sít' IRC), ale průlom v netechnické veřejnosti zaznamenal až v roce 1996, po vzniku programu ICQ (Mirabilis). E-mail v dobách vytáčeného připojení znamenal rychlý způsob komunikace. V dnešní době, kdy takovýto typ internetového připojení je v civilizovaném světě téměř zapomenut, již není dostatečně rychlý. Lidé chtějí mít možnost diskutovat se svými blízkými stejně rychle a jednoduše, jako kdyby s nimi zrovna hovořili, i když jim to například fyzická vzdálenost zrovna neumožňuje. A protože telefonní hovor není vždy vhodný, IM se stal okamžitě velice oblíbený. Umožňuje vytvořit seznam přátel, u kterých zobrazuje informaci, zda je daná osoba zrovna připojena, či ne, zasílat emoji, webové linky, části zdrojových kódů a v moderní době i video a soubory, případně s jiným uživatelem hrát různé hry. V poslední době se také IM rozšířil do našich mobilních zařízení a částečně vytlačuje dříve velice oblíbené SMS zprávy. Výhody IM komunikace objevil i profesionální svět a je tedy využívána i v podnikových prostředích. Možnost nastavit si určitý stav uživatele (dostupný a kde, pryč, krátkodobě nedostupný, nerušit, jednání) je v této sféře taky hojně využívána.

[11]

5.2 Protokoly pro \“instant Messaging\“

5.2.1 Jabber (XMPP)

Vznikl v roce 1998 autorem Jeremie Miller. Veřejně dostupná verze se objevila v roce 2000. Po uznání protokolu jako standard, byl přejmenován na XMPP. Klienti Jabberu nekomunikují přímo spolu, ale přes server, když je sít' decentralizovaná, neexistuje jeden hlavní server s jedinou databází uživatelů. Server si může vytvořit i uživatel sám a přesto nepřichází o možnost komunikace s uživateli jiných Jabber serverů, stačí se připojit

klientem na jiný server. Uživatelské jméno ve tvaru jmeno@server pak na první pohled může říct domovský server, kde je uživatel zaregistrován. Z původního protokolu pro „lehce šílené“ uživatele Linuxu, se stal Jabber pro veřejnost důvěryhodným po tom, co ho použila internetová firma Google a automaticky svým uživatelům vytváří uživatelský účet, hned při registraci e-mailových služeb. Nepopsatelné plus navíc představují neexistující licenční podmínky, síť je tedy možné použít k čemukoli a jakkoli, bez rizik.[35]

5.2.2 ICQ (I Seek You)

Jako projekt uveden v roce 1996 firmou Mirabilis, vzhledem k okamžitě rostoucí popularitě sítě postavené na protokolu OSCAR, firmu Mirabilis po dvou letech provozu koupila firma AOL. Tato síť je provozována s jedním centrálním serverem, s informacemi o uživateli, kde každý uživatel má unikátní číslo UIN sloužící k přihlášení. ICQ má licenční a provozní podmínky, se kterými musíte při registraci souhlasit, síť se tak nesmí používat ke komerčním účelům, nesmí ji používat děti mladší 13-ti let, nesmí se do ní přistupovat jinak, než oficiálním klientem, či si firma AOL nárokuje autorská práva na cokoli, co přes ICQ zašlete. Dalšími úskalími jsou zobrazovaná reklama v oficiálním klientovi, nebo zpomalení operačního systému nesmyslnými nároky oficiálního klienta. Přesto po dlouhou dobu, zvláště v České Republice, si veřejnost pod pojmem IM představovala právě ICQ a to ve značné míře nejspíše špatnou obecnou informovaností veřejnosti, či díky přidané hodnotě možnosti využití webového klienta a přidružených zábavných služeb, jako logické hry lehce spustitelnými mezi komunikujícími. [35]

5.2.3 GAGU

Spíše okrajová síť, masivně používaná v Polsku, kde i vznikla. Je sponzorovaná reklamou.[35]

5.2.4 MSNP

Síť vytvořená Microsoftem. Až do nedávna masivně, díky integraci klienta přímo do OS Windows a z toho vzniklé velké základně uživatelů (počet registrovaných členů v roce 2008 byl 452 milionů). S postupným úpadkem přišla řada modifikací klienta i sítě, nyní se jmenuje Windows Live a byla spojena s dalšími službami, aby uživatele přilákala zpět. [35]

5.2.5 GoogleTalk, GTalk (XMPP)

Jak bylo řečeno v sekci 5.2.1, protokol XMPP provozovaný na vlastních serverech společností Google. Součástí služby Gmail a postupně prochází transformací klienta s pravděpodobným vzniknutím vlastního protokolu. [35]

5.2.6 Skype

Šifrovaná síť s neodhaleným protokolem, původně zaměřena hlavně na hlasovou komunikaci přes internet, která vnikla v roce 2003. Postupně zavedla možnost zasílání zpráv mezi uživateli. Uživatele sdružuje na jednom serveru. Při registraci je nutné souhlasit s podmínkami služby a ty zaručují firmě využívání klientského počítače jako komunikačního uzlu, přes uživatelské internetové připojení tak můžou proudit data, pro toho uživatele absolutně nepotřebná, akorát zvyšující provoz na lince a zatěžující hardware.[35]

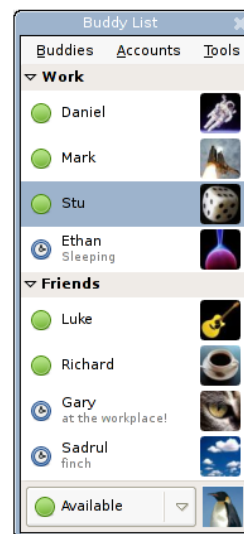
5.3 Klienti a služby pro \“instant Messaging\“

5.3.1 Pidgin

Multiplatfomní a uživatelsky přívětivý klient s podporou mnoha protokolů. [28]

Mezi ty hlavní patří: XMPP, Google Talk, AIM, ICQ, MSN, IRC

Podporuje přenosy souborů, uživatelské statusy, změnu emoikon. Je distribuován zdarma a bez reklamy a podporuje mnoho jazyků.



Pidgin IM Obrázek 5-1

5.3.2 Skype

Hybridní IM klient zaměřený hlavně na přenos hlasu a tím spojené moderní odnože, přenos videa a utváření videokonferencí. S rozsáhlou uživatelskou základnou se stal tak zajímavým, že byl koupen Microsoftem.[3]

S umožněním zasílání psaných zpráv přinesl podporu i přenosu souborů a propojení s uživatelským účtem služby Facebook.



Prostředí IM Skype Obrázek 5-2

5.3.3 Windows live messenger

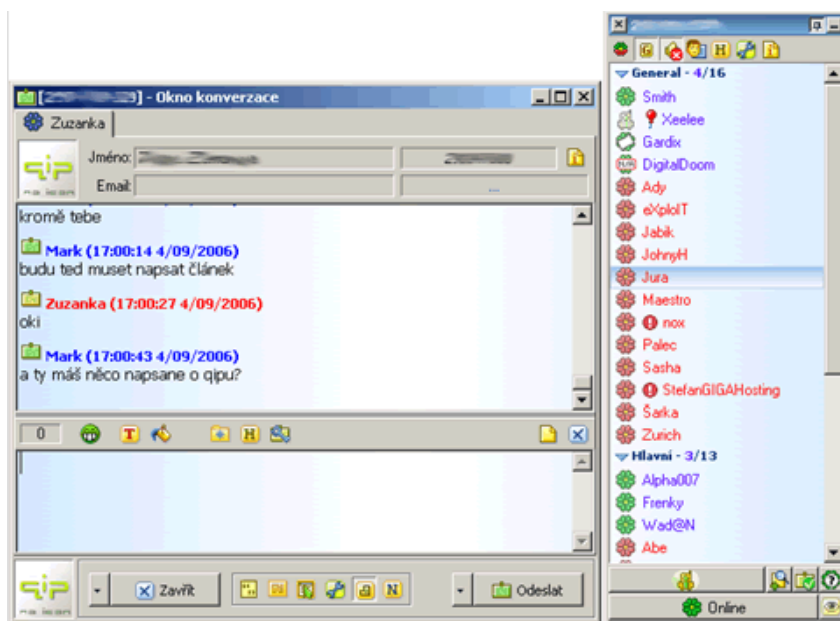
Dříve znám pod jménem MSN Messenger. Klient se nacházel v každém OS Windows. Nyní po akvizici Skype Technologies došlo ke sloučení účtů a původní verze je již provozována jen v Číně. [39]



Windows Live Messenger Obrázek 5-3

5.3.4 QIP

Velice oblíbený klient pro protokol ICQ a nově XMPP. Klienti jsou dostupní pro OS Windows a pro přenosná zařízení. Oproti originálnímu ICQ klientu je velmi úsporný co se týče zdrojů, nezatěžuje tolik OS a také nezobrazuje reklamu. [31]



Okno zprávy a seznam kontaktů IM QIP. Obrázek 5-4

5.3.5 ICQ

Domácí klient pro protokol OSCAR. V aktuální verzi 8 nabízí krom podpory zaslání zpráv volné videohovory, volné hlasové hovory, volání na pevné i mobilní linky, sociální sítě, sdílení souborů, upozornění na email, emoikony, motivy, nálady. [13]



IM klient ICQ. Obrázek 5-5

5.3.6 Facebook

Sociální síť, která stojí za ústupem klasických IM. Umožňuje zaslání zpráv přímo v prohlížeči mezi uživateli, po tom, co se spojí jako přátelé. Komunikace probíhá přes XMPP protokol a pro OS Windows Facebook nabízí i klasického IM klienta.[23]



IM klient
Facebook pro
OS Windows.
Obrázek 5-6

5.3.7 Miranda

Klient pro OS Microsoft Windows, open-source a podporuje mnoho protokolů.

Výhodou je vysoká modulárnost, nevýhodou složitost nastavení a častá nestabilita aplikace. [24]



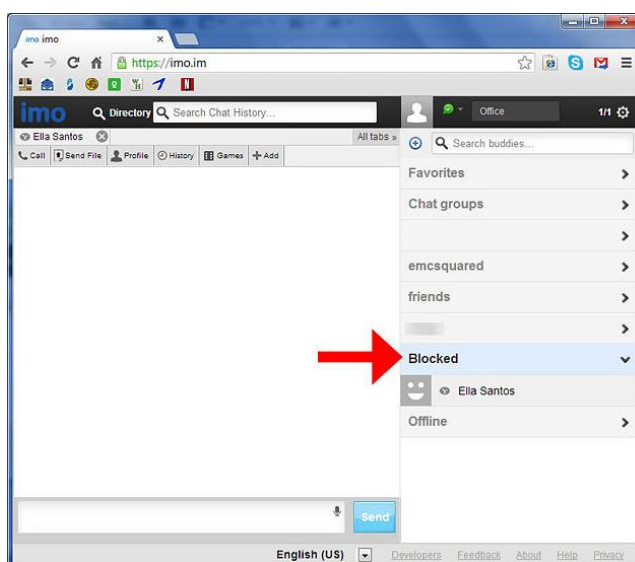
Miranda IM Obrázek 5-7

5.3.8 Imo.im

Imo je profesionálně vyvíjená webová služba pro IM komunikaci vše v jednom. [15]

Umožňuje připojení na Facebook, Google Talk, Yahoo, AIM, ICQ, MSN.

Z toho je vyplývající podpora protokolů XMPP a OSCAR.



Web IM IMO. Obrázek 5-8

5.3.9 Gtalk

Desktopový klient je označen přívlastkem testovací veze, ač je plně funkční. Podporuje pouze XMPP protokol a na rozdíl od webového klienta nedokáže přenášet video. Zato dokáže ukládat konverzaci do samostatné složky v Gmailu.[30]



Desktopový klient GTalk. Obrázek 5-9

II. PRAKTICKÁ ČÁST

Při mapování aktuálního užívání OTR, jsem si všimnul převážného zaměření na stand alone komunikátory, tedy klienty, které se spustí jako samostatný program operačního systému. Aktuální trend přesouvání veškeré práce do prohlížeče, užívání webových služeb s přidruženou instant messaging komunikací a snižujícího se soukromí, stál za vnuknutí myšlenky využít OTR protokol v IM komunikátoru, běžícího na webové stránce vlastní domény. Takovýto komunikátor, zaměřený na malý počet uživatelů, by představoval nástroj pro soukromou komunikaci, navíc mimo velké hráče na internetu, tedy s dalším přidruženým prvkem soukromí. Vyhnutí se komunikace přes servery velkých hráčů zabraňuje obavám z občasných spekulacích teorií a informacím, že dané firmy své servery využívají ke sledování práce uživatelů a jejich komunikace. A také samozřejmě zvýšení komfortu uživatele, snížením zobrazované reklamy k nule na webu komunikátoru, je v dnešní době příjemný a neobvyklý prvek.

Komunikátor by měl umožňovat uživatelům přihlášení, automatické nalogování do hlavní chatovací místnosti se zobrazením připojených uživatelů. Možnost jak nešifrované, tak šifrované komunikace. Zasílání zpráv s textem, zprovoznění a možnost zasílání souborů či hlasu, ať už v nešifrované, nebo šifrované podobě až jako přidanou hodnotu po delším zkušebním provozu. Takováto služba by měla přinést uživatelům možnost soukromé komunikaci i na sdílených počítačových stanicích s omezenými instalačními právy.

Došlo k volbě protokolu, přes který bude celá komunikace probíhat. Pro otevřenost a rozličné možnosti použití byl zvolen Jabber, tedy XMPP.

Využita byla OTR knihovna psaná v jazyku JavaScript, IM komunikátor Candy IM psaný také v jazyku JavaScript. Pro implementaci OTR knihovny byl použit český program PSPad, testování bylo prováděno v internetovém prohlížeči Google Chrome, s pomocí nástrojů pro developery, jako odladování javascriptů a konzole pro sledování událostí. Pro zprovoznění IM komunikátoru byl zvolen lokální http server project Apache, pro Jabber server, lokální instalace Openfire.

Pro projekt zakoupená doména a web hosting nemohly být ve výsledku použity. Problém se objevil při vytváření spojení na Jabber server, XMPP protokol je sice hostingem podporován, ale ve výsledku pouze k vytvoření účtu na jejich serveru (využití zřízené e-mailové schránky k doméně a její adresy), další nastal při konfiguraci Apache serveru, kdy parametr http-bind pro BOSH (http-binding) service není umožněno provozovatelem

změnit, zavedení podpory ani do budoucna neuvažují, jak zněla odpověď technické podpory na dotaz spuštění této služby.

6 PROŠŘEDKY A PRVKY K IMPLEMENTACI

Bylo nutné zvolení IM komunikátoru a knihovny, na který proběhne implementace. Tato kapitola je představí podrobněji.

IM komunikátor Candy byl vybrán pro obsáhlou dokumentaci, vysokou možnost úprav a podporu protokolu XMPP

6.1 IM komunikátor Candy

Komunikátor byl vytvořen pány Michael Weibel a Patrick Stadler. [4]

Je založen na programovacím jazyku JavaScript, zaměřen na více uživatelů v reálném čase, podporuje rozšíření a úpravy již definovaných událostí, pracuje se všemi hlavními internetovými prohlížeči a byl stvořen pro Jabber (XMPP) protokol s pomocí standardu XEP-124. Díky tomu nevznikají další nároky na straně uživatele, jen mít nainstalovaný webový prohlížeč.

K vytvoření Candy IM bylo využito těchto knihoven:

- **Strophe.js** XMPP knihovna pro JavaScript
- **jQuery** write less, do more.
- **mustache.js** knihovna pro zjednodušené vytváření HTML šablon

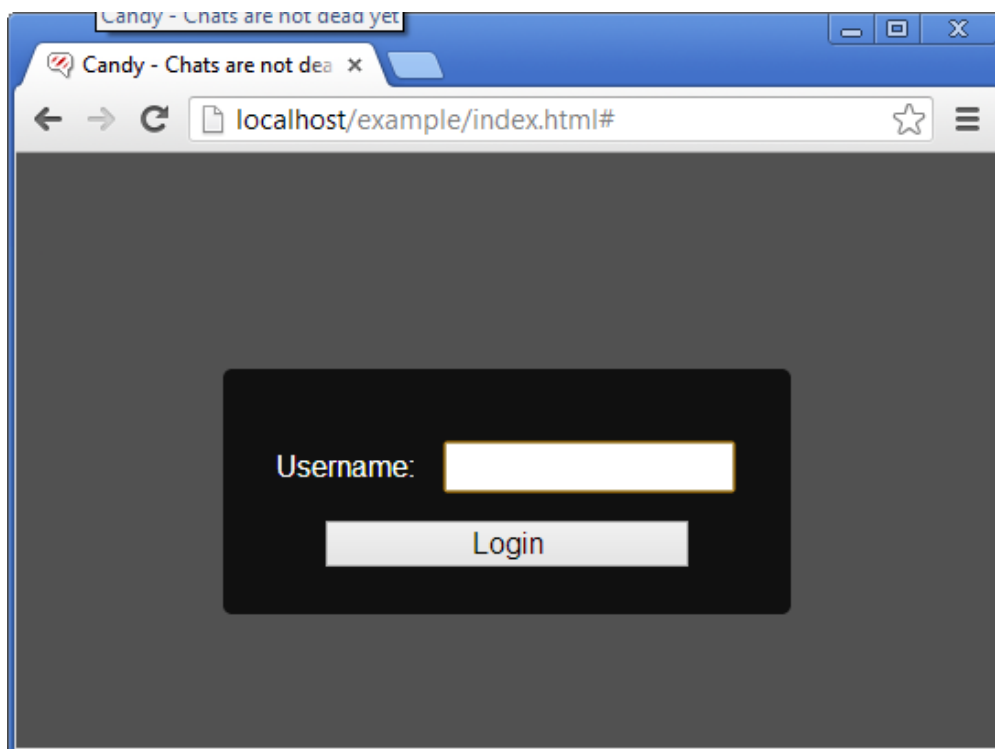
Candy IM ke spuštění potřebuje:

- Přístup na Jabber server s povoleným http-binding parametrem
- HTTP server s možností nastavení proxy parametrů pro cross-domain AJAX požadavky
- Nastavení HTTP serveru s načtenými moduly `mod_rewrite`, `mod_proxy`, `mod_proxy_http`.
- Možnost nastavit na HTTP serveru parametr `http-bind` naslouchacího portu.

Po splnění těchto podmínek je možno spustit samotný komunikátor, pomocí přímo přiloženého příkladu uloženého ve složce `examples`, souborem `index.html`

Pro nalogování do klienta může být využito několika metod:

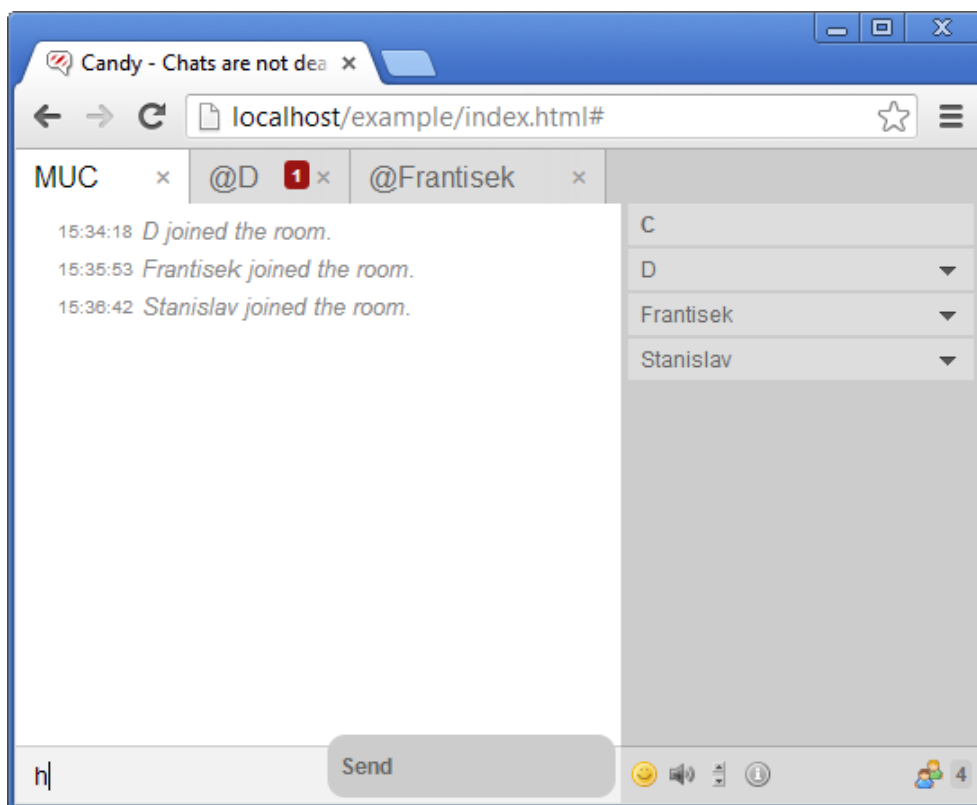
- JabberID a heslo
- Předem definované JabberID a heslo
- Nick – přezdívka volená přímo při logování (nutnost použití spojení na Jabber server, který tento typ logování umožňuje)
- Předem definovaný Nick
- Předem definované JabberID, uživatel je tázán pouze na heslo



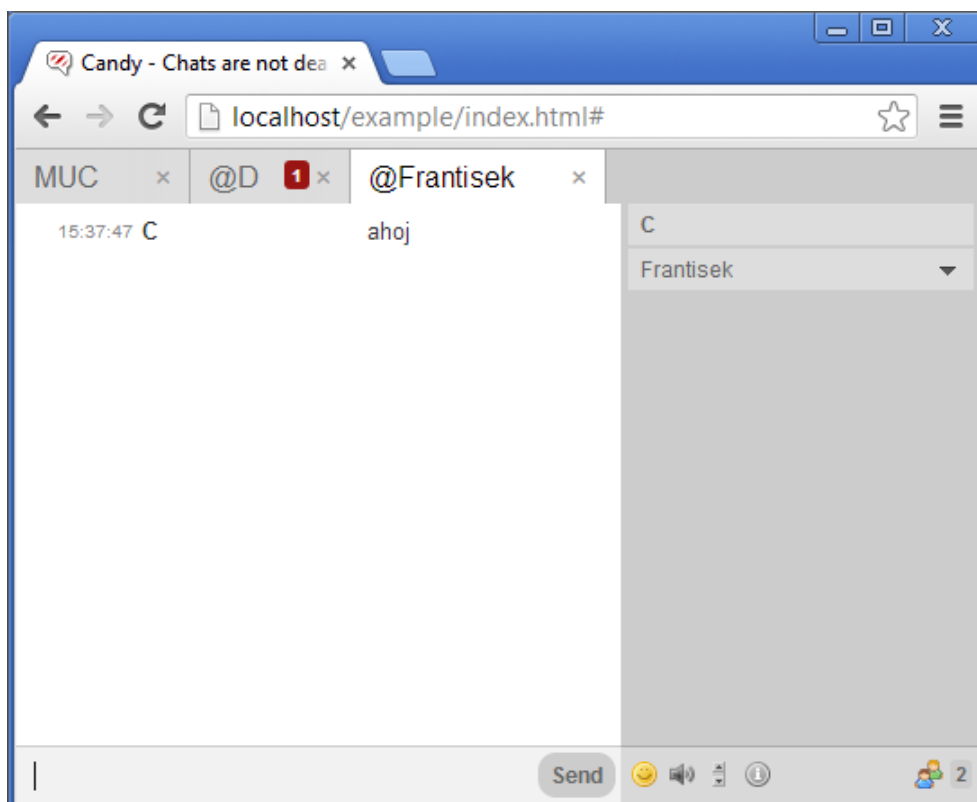
Logovací formulář pro vstup do Candy IM (metoda Nick, bez hesla). Obrázek 6-1

Po nalogování dojde k připojení uživatele do hlavní chatovací místnosti, pokud je na Jabber serveru místnost nastavena a zároveň ji má i Candy IM definovanou.

Uživatel může komunikovat se všemi uživateli dohromady, nebo s některými samostatně přes privátní okna. Situace je vyobrazena na následujících obrázcích.



Společná místnost pro chat s listem dostupných uživatelů. Obrázek 6-2



Okno privátního chatu. Obrázek 6-3

6.2 OTR knihovna

Knihovna byla vytvořena panem Arlo Breault a publikována na serveru GitHub.[1]

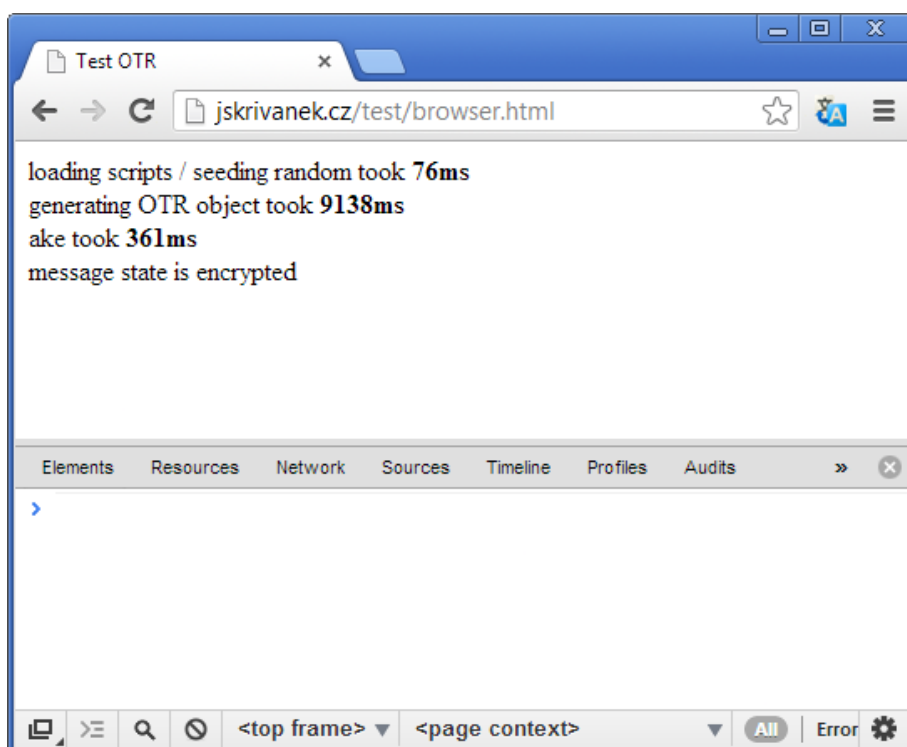
Knihovna je napsána v jazyku JavaScript a pro její stvoření a funkčnost OTR protokolu bylo využito ještě dalších knihoven:

- crypto-js – knihovna s implementací šifrovacích a hashovacích algoritmů (SHA, HMAC, AES)
- bigint.js – knihovna s výpočetními operacemi používanými v kryptografii
- salsa20.js – knihovna obsahující proudovou šifru, založenou na pseudonáhodné funkci
- eventemitter.js – knihovna systému událostí pro jQuery

Definovány jsou verze OTR protokol 2 a 3.

Součástí dokumentace ke knihovně je i testovací procedura pro ověření funkčnosti knihovny. Procedura je spustitelná v prohlížeči a to spuštěním browser.html, pokud by se v ní nacházela chyba, byla by vypsána do konzole nástrojů pro vývojáře.

Stav funkčnosti je viditelný na následujícím obrázku.



Test funkčnosti OTR knihovny. Obrázek 6-4

6.3 Apache http server project

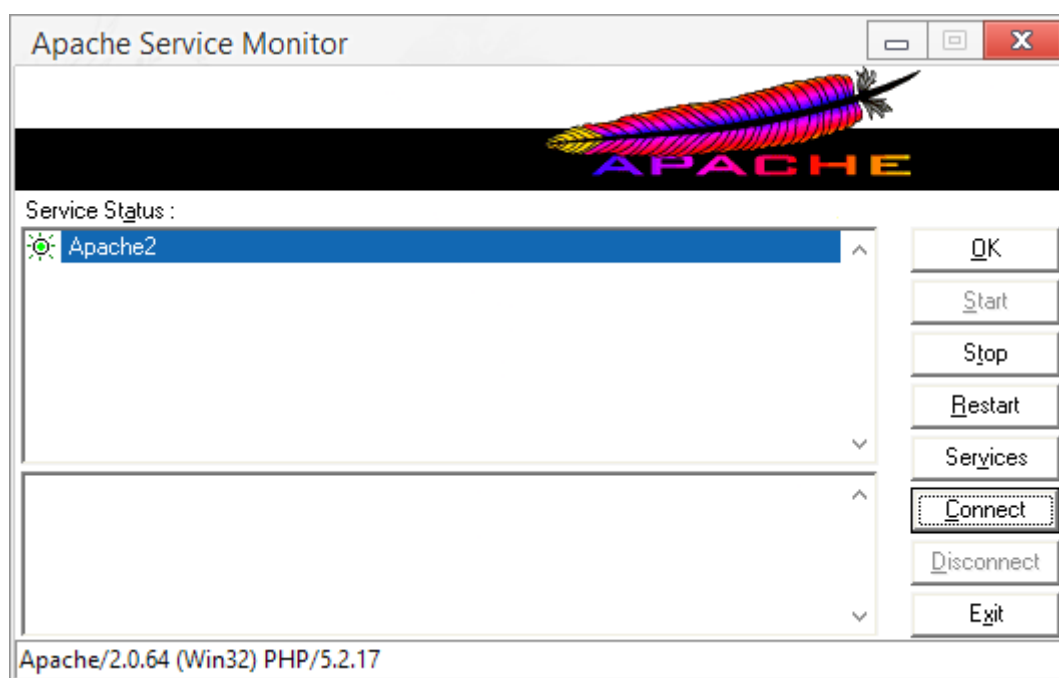
Pro potřeby lokálního HTTP serveru byl zvolen doporučovaný Apache, ve verzi 2.0.

Apache http server je nejoblíbenějším softwarem na internetu od roku 1996.[38]

Bylo nutné povolit následující moduly:

- mod_rewrite
- mod_proxy
- mod_proxy_http.

A také nastavit http-bind url pro zmiňovaný BOSH service. Toto nastavení bylo provedeno v souboru .htaccess společně s jeho umístěním do složky s index.html souborem, ze kterého docházelo ke spouštění Candy IM.



Servisní okno Apache http serveru. Obrázek 6-5

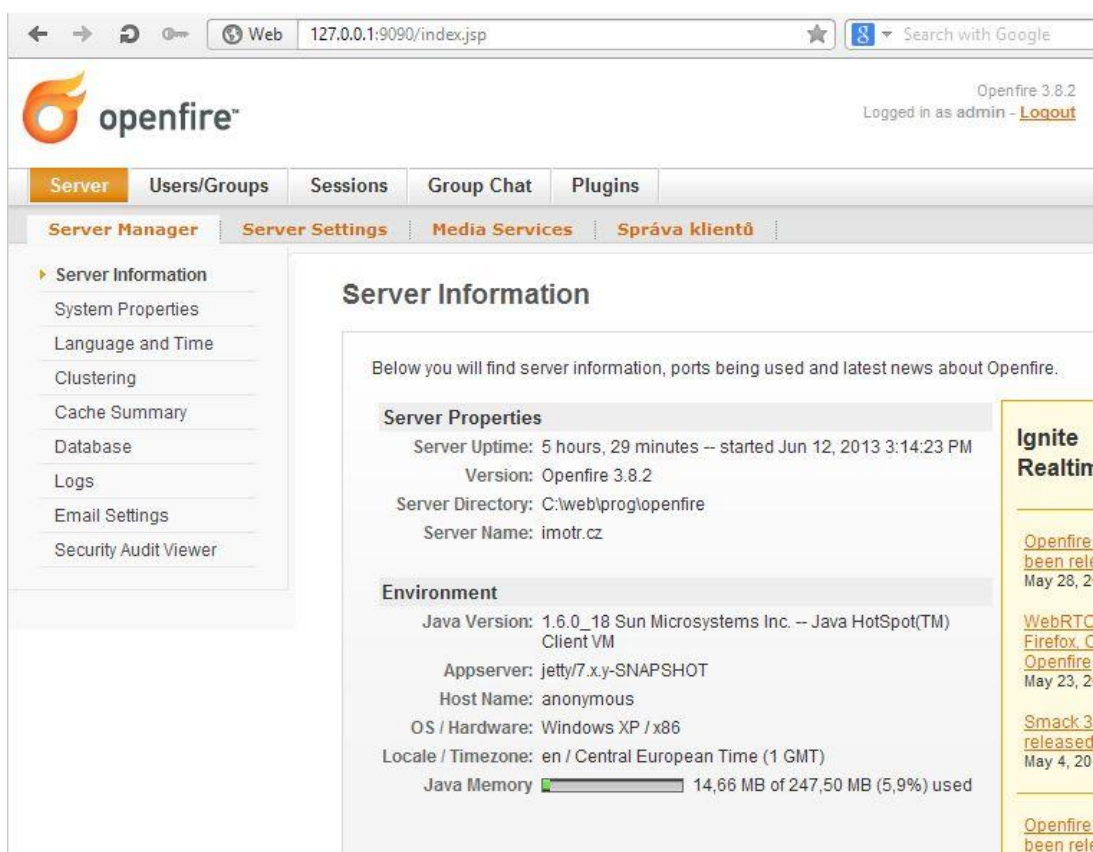
6.4 Openfire

Software umožňující komunikaci v reálném čase a to jak v instalaci pro lokální použití, tak v instalaci přímo na OS serveru. Založen na jediném, široce používaném, otevřeném protokolu Jabber (XMPP). [14]

Tímto softwarem bylo docíleno vytvoření lokálního Jabber serveru s možností vlastní definice a jak prostředků pro IM, tak nastavení síťových portů pro spojení ze serveru Apache. Tedy již zmiňovaný http-bind parametr.

Bylo nutné nastavení těchto parametrů:

- Jméno serveru
- Povolení HTTP Bind parametru a definice jeho portu
- Aktivace pluginů Client Control a Search
- Jméno chatovací místnosti
- Conference.



Uživatelské rozhraní pro konfiguraci Openfire. Obrázek 6-6

7 IMPLEMENTACE OTR KNIHOVNY NA IM CANDY A OVĚŘENÍ FUNKČNOSTI

Kapitola se bude zabývat praktickým řešením teoretického návrhu popsaného v úvodu praktické části a ukáže průběh OTR komunikace mezi dvěma uživateli.

7.1 Implementace OTR knihovny

Candy IM je spouštěno z HTML statické stránky, načtení a spuštění knihoven má tedy na starost webový prohlížeč. Implementace knihoven OTR protokolu proběhne na stejné místo. Defaultní implementaci popisuje dokumentace.[1]

Candy IM je navržen i s možností různých úprav. Pro tyto úpravy podle dokumentace slouží předdefinované Event Hooks, ty jsou volány před nebo po provedení určitých událostí (Event).[4]

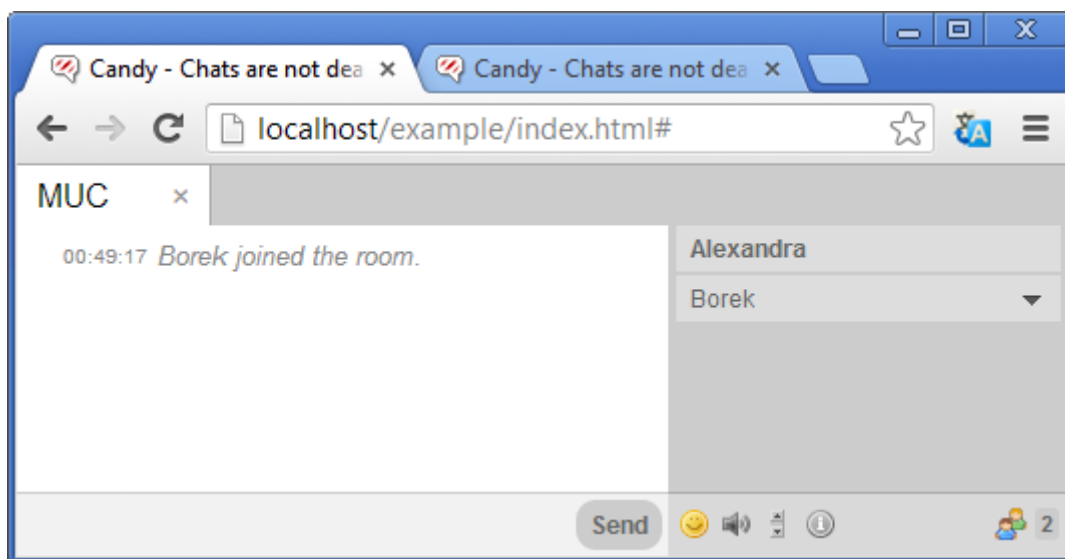
Po prostudování a otestování OTR knihoven došlo ke zjištění, že jich nepůjde plně využít, funkce OTR protokolu se volaly a uskutečňovaly až po provedení eventu, tedy pozdě.

Další krok tedy vedl k upravení samotných událostí přímo v knihovnách Candy IM, to ovšem vedlo také do slepé uličky, Candy IM má knihovny po kompresi a dokumentace přesně nepopisuje jak je upravit.

K implementaci došlo na úrovni pozměnění jádra IM komunikátoru při inicializaci z HTML stránky. Na této úrovni měly být definovány pouze Event Hooks a ne změny celých událostí. Proto Candy IM nedokáže využít všech možností OTR protokolu verze 2.

7.2 Test Candy IM s OTR

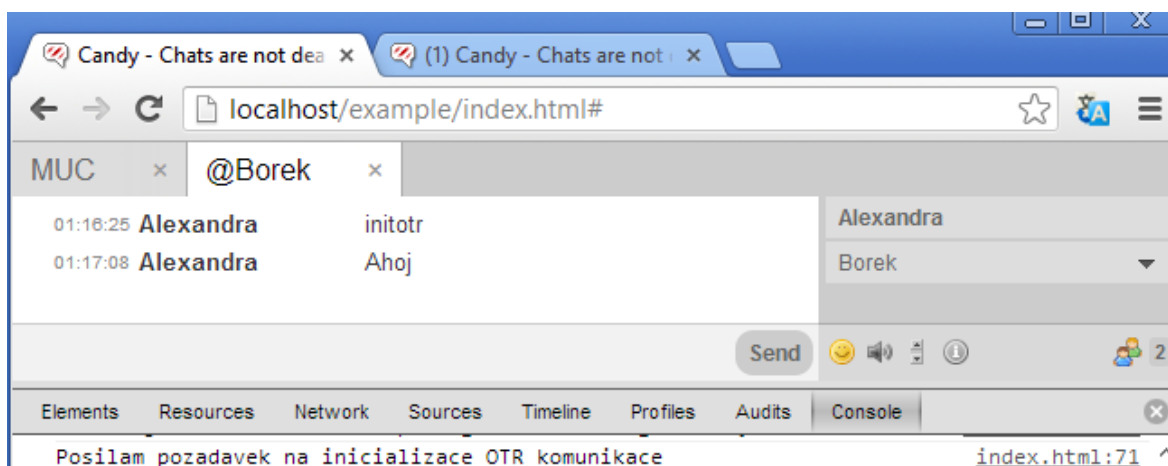
Pro otestování museli být zavedeni dva uživatelé. Alexandra (A), Borek (B).



Dva aktivní uživatelé v Candy IM. Obrázek 7-1

K inicializaci OTR komunikace dojde po tom, co jeden z uživatelů zašle soukromou zprávu jinému uživateli ve tvaru initotr (initiate OTR) a dojde tak k autentizaci účastníků.

To by mělo zůstat uživatelům skryto, výpisy jsou proto pro demonstraci zvoleny do konzole.



Inicializace OTR komunikace zasláním initotr Obrázek 7-2

Uživateli je jeho dlouhodobý klíč přiřazen hned po nalogování do Candy IM. Generace klíče je náročná, proto je vhodné vygenerovat klíč dopředu.

Inicializační zprávou zašle A dotaz B, zda OTR podporuje

```
Posílám zasifrovanou zprávu: ?OTRv23? index.html:54
SENT: <body rid='795396892' xmlns='http://jabber.org/protocol/httpbind'
sid='65c85ef5'><message to='muc@conference.imotr.cz/Borek'
from='65c85ef5@imotr.cz/65c85ef5' type='chat' id='795' xmlns='jabber:client'><body
xmlns='jabber:client'>?OTRv23?</body><x xmlns='jabber:x:event'><composing/></x>
</message></body> candy.min.js:1
RECV: <body xmlns='http://jabber.org/protocol/httpbind' ack='795396892' />
```

Obsah inicializační zprávy Obrázek 7-3

Příjem této zprávy na straně B a odpověď, která spouští autentizující proces

```
RECV: <body xmlns='http://jabber.org/protocol/httpbind'><message
xmlns='jabber:client' to='6d2fdf47@imotr.cz/6d2fdf47'
from='muc@conference.imotr.cz/Alexandra' type='chat' id='795'><body>?OTRv23?</body><x
xmlns='jabber:x:event'><composing/></x></message></body> candy.min.js:1
[Jabber] Message candy.min.js:1
[Jabber:Room] Message candy.min.js:1
Přijímám zprávu: ?OTRv23? index.html:123
d-h commit message otr.min.js:11
instance tags otr.min.js:11
Posílám zasifrovanou zprávu: ?OTR|54fa30c2|c1ca99d2,1,3,?
OTR:AAMCVPowwsHKmdIAAADEVbdsx5xzX0f2UTFGpO5LybN1VHD180Rgy43GYIYrAvHKKMpDa3T5VoyyH1Wse
DmtdtdsNuH6kGegTU101z1Ify4nwFCV7CGRJkhYBUDfTeAhQ1Cnc36, index.html:105
SENT: <body rid='1548161223' xmlns='http://jabber.org/protocol/httpbind'
sid='6d2fdf47'><message to='muc@conference.imotr.cz/Alexandra'
from='6d2fdf47@imotr.cz/6d2fdf47' type='chat' id='9678' xmlns='jabber:client'><body
xmlns='jabber:client'>?OTR|54fa30c2|c1ca99d2,1,3,?
OTR:AAMCVPowwsHKmdIAAADEVbdsx5xzX0f2UTFGpO5LybN1VHD180Rgy43GYIYrAvHKKMpDa3T5VoyyH1Wse
DmtdtdsNuH6kGegTU101z1Ify4nwFCV7CGRJkhYBUDfTeAhQ1Cnc36,</body><x
xmlns='jabber:x:event'><composing/></x></message></body> candy.min.js:1
Posílám zasifrovanou zprávu: ?OTR|54fa30c2|c1ca99d2,1,3,?
```

B tedy spouští AKE (viz sekce 3.2.1)

Dále bude komunikace sledována již jen na straně Alexandry,

```
Přijímám zprávu: ?
OTR|54fa30c2|c1ca99d2,2,3,4iVezE53hf3TyjKuOPPe32MHCz6eSkfyt3kv/gQzGgSQOLeba+v1jH70chq
jYy88zY+GMFmOyktHKZ50ihMMMTdTWB0C4PsV9DcLZw4cDaD7aAq4nmHDEL8WQr+SYp7DSN8w2gsm3vXI,
index.html:123
```

```
Přijímám zprávu: ?
OTR|54fa30c2|c1ca99d2,3,3,OSB9o1gAAACCZw10Adj5VVJLFzWuNdkT+2MD5U47GVzYdCLL9JkQtRA==.,
index.html:123
```

D-H vytvoření sdíleného tajemství (viz sekce 3.2.1)

```
a-n key message otr.min.js:11
instance tags otr.min.js:11
Posílám zasifrovanou zprávu: ?OTR|c1ca99d2|54fa30c2,1,3,?
OTR:AAMKwcqZ01T6MMIAAADAAzeZTWfkQpI90sXDRY2WnN3t2Uh1nEguwOKuClJjECdo/XLI6uy85+Bmyz5qk
KpzzpSERzRwRb6h3YeBX9Qgerf2n7FfjFK5F63UkKEW58RP1R2PYxq, index.html:54

Posílám zasifrovanou zprávu: ?
OTR|c1ca99d2|54fa30c2,2,3,vOL00eMQW1n1cdMF15XRqIabZfDj0GQUVWEx+1caBTUckBgEhXcr8CkDJUn
hhXUQblvXF0ZhFWtb7xEBy09QK1yKDnQ00aaATS3tjow7unAzZUJ8fxZrOEmdBaZtbFeh+f9StrEtBJTj,
index.html:54

Posílám zasifrovanou zprávu: ?OTR|c1ca99d2|54fa30c2,3,3,W., index.html:54
```

Prijimam zpravu: ?OTR|54fa30c2|c1ca99d2,1,6,?
 OTR:AAMRVPowwsHKmdIAAAQ9X8wqB3VSoewZqZmSVUekAAAdKXqtnOnpceM8E2FjD9fpOjQRszS+UuC7h7Y
 Eq7fsAYCFHwykCkT0o9zNKIysevU2IG/hTnUi8KtsKkU5IIj11nF3R,
[index.html:123](#)

Prijimam zpravu: ?
 OTR|54fa30c2|c1ca99d2,2,6,gMVdo56z9e5u3wk1FTDqd0WrkzrAxDzurERdh1nbFalpDwQDfxZAAe4Kmaz
 S03KTWClr1EMuZ9gwLnfbTqBGGrK9hF7ZW9vDXIOd1VTaVEe5k9GoN/G73gMWPVw/fhi0jTTRwtUFDqfq,
[index.html:123](#)

Prijimam zpravu: ?
 OTR|54fa30c2|c1ca99d2,3,6,C8t/cYtYUueB2UaIciBNiOgYKPEK14BwuH+tID01pEJ8ndyuwgr611TUB7A
 0A49MuJncBRkIc0E0z3SaE0oGRciTym6jd+y1ZT1d/kgyjYKiNmKfJhnvoM5VLRb9+OgbKOQtclNSxsY,

Prijimam zpravu: ?
 OTR|54fa30c2|c1ca99d2,4,6,PvmpecgbaYG4eseOwoKNVGBGixfJB5ItCd1z2FwxVktKzM3L47GQ9gIDn5L8
 gZglh9sEp/ipRET0WoB92VZrvBBmv/YLA91tcWqp70ZUeBsYIIjWMw21eCYU1YaE7jFY5e/2XplVDS0yA,
[index.html:123](#)

Prijimam zpravu: ?
 OTR|54fa30c2|c1ca99d2,5,6,pVS2Se5pPDax71Q1k/RseGBVqIFXXecB09f4+29NEjFGD6j6KkXFXOM7XYx
 zAqo3e8pI32HNJ9aCR8J/byxp819k9OiiTtY4k+Tv/seEGHtGigSH8Ckb+up6xvhZ1iQBfteLSy+T4i5F0,
[index.html:123](#)

Ověření šifrovaného kanálu a stran (viz sekce 3.2.1)

Prijimam zpravu: ?OTR 54fa30c2 c1ca99d2,6,6,=.,	index.html:123
signature message	otr.min.js:11
instance tags	otr.min.js:11
success	otr.min.js:11

Posílám zasifrovanou zpravu: ?OTR|c1ca99d2|54fa30c2,1,5,?
 OTR:AAMSwcQZ01T6MMIAAHSzpUbwL7wjX13CcnghxwhdaVIIIZwpwHhXmvtL3AT1FwsIj1Jgk6eP9S44iWi
 CkyhH7HBEhLO19DjOOvq8NH9WfGn+1luF2CblosR3U6SND0V75dv3n,
[index.html:54](#)

Posílám zasifrovanou zpravu: ?
 OTR|c1ca99d2|54fa30c2,2,5,VP5KO2JNL16SdOh+mGSlymtMCO2Yg4a1S2YdayCim179ouOWLXyv+JXGA66
 pT1HHKJWRfDIXeYGPj1GTPqy5cBKwHGHuSYCfxIbWSZrAR1DHM11XfMgC247no7u1PW0mfOZTYzE23Ddi,
[index.html:54](#)

Posílám zasifrovanou zpravu: ?
 OTR|c1ca99d2|54fa30c2,3,5,n+jNzTrc9a8kmX5TOU78rm/wJvRrrjb9Cz2kfWiIBZPS/wOgju1ZUJC1S1S
 jYf1H4EpBgM3pYjQZhm9pTh7JqQM7Vq0zoX4w1YC75ZCUEKs/Bnp/0w8U81HiSqGdPyxNO81cs1RKioBV,
[index.html:54](#)

Posílám zasifrovanou zpravu: ?
 OTR|c1ca99d2|54fa30c2,4,5,IvacFZV8dsE7yYIORVdvzfozDX0JznfZuixRqW5xci3Xk6w12iVyPPSP7y
 obFw8+m0Meel7sL0aONdp1IGbVgUFBWk/1+2sQLEPkHteyn/xIJr2UEeWcMbBnz9Cs8zHz/4nltz85X8e,
[index.html:54](#)

Posílám zasifrovanou zpravu: ?
 OTR|c1ca99d2|54fa30c2,5,5,N1UzzPsO+jnc+hciUc5zE8QwnhDxQpFyRQ71J0mZuazy64VvfiLACcT1u7L
 qdcJiyfIBZmjQ8mTx/C/oFR4VEA/29HibgTD1scfugeIQQGpi/00aP.,
[index.html:54](#)

Autentizace a ověření uživatelů proběhlo úspěšně. Šifrovaný kanál zaveden, může dojít k přenosu datových zpráv

Po ověření živitelů již je možné přenést i zprávu:

Situace vypadá následovně:

Vložení zprávy, šifrování a zaslání A

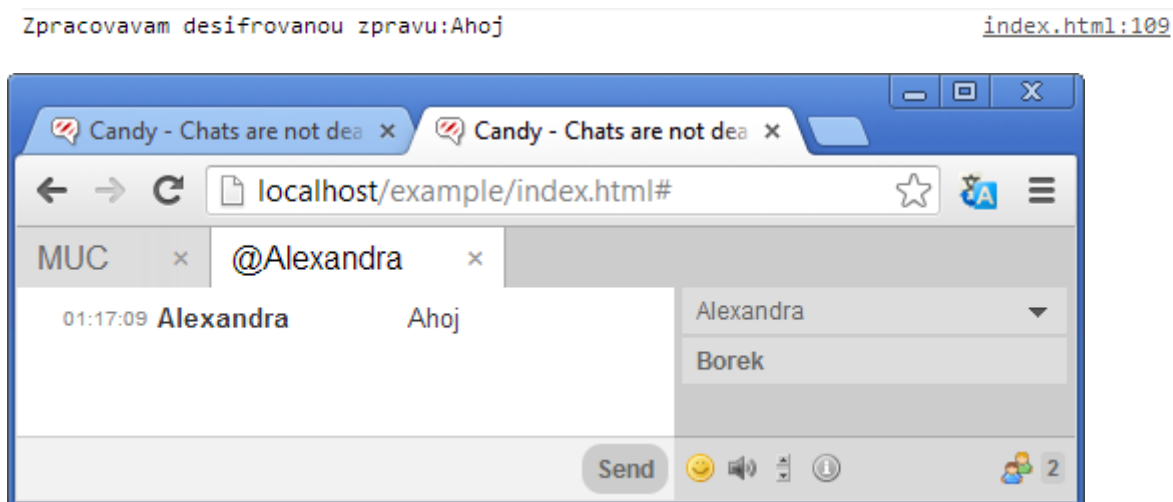
```
Posílám zprávu:Ahoj index.html:77
Posílám zasifrovanou zprávu:?OTR|c1ca99d2|54fa30c2,1,3,?
OTR:AAMDwcqZ01T6MMIAAAAAQAAAAEAAADAWAxS761ubkVePgBoJ1GTMRsKIvfzG5jm7dGu4z7xDQkJDvWx
UYdglJOaORc8KPXt1g4xZey/66onwdAHWfMbvdhMCNrA1BrM1RYhQN, index.html:54

Posílám zasifrovanou zprávu:?
OTR|c1ca99d2|54fa30c2,2,3,H9hjzDBqEzMxIr32UIBusLVZXqzGzz9PDVYTCd4fUWX50wqpX0hMo5o6iB
GfPMj8+v1QiChcuM19Ro9/06Msj+KL2iSmuAFYK2sqvh5cKDmAqdrOz+HhKfvscvxLGODwSeFj71EhYDC,
index.html:54

Posílám zasifrovanou zprávu:?
OTR|c1ca99d2|54fa30c2,2,3,H9hjzDBqEzMxIr32UIBusLVZXqzGzz9PDVYTCd4fUWX50wqpX0hMo5o6iB
GfPMj8+v1QiChcuM19Ro9/06Msj+KL2iSmuAFYK2sqvh5cKDmAqdrOz+HhKfvscvxLGODwSeFj71EhYDC,
index.html:54

Posílám zasifrovanou zprávu:?
OTR|c1ca99d2|54fa30c2,3,3,MtFzVY61YNnGsAAAAAEEAAAAEhI2G963psqzjE2ytrtM0/U6vJZMFWL
PpAAAAA=., index.html:54
```

Strana B získá zprávu:

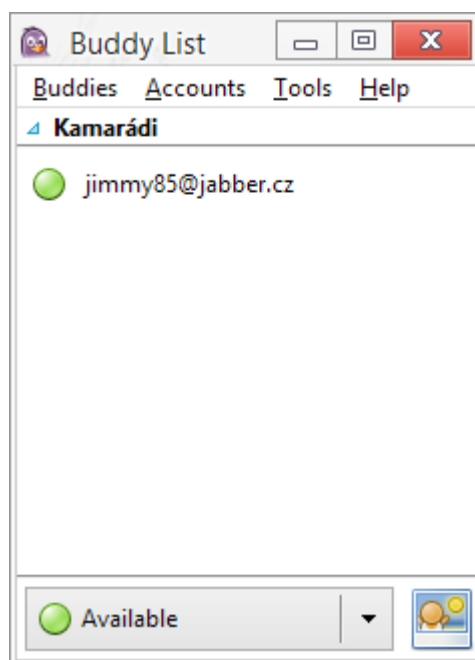


Zde je vidět, že zpráva od uživatele A přišla uživateli B a byla dešifrována.

7.3 Hotové řešení OTR plug-in pro Pidgin

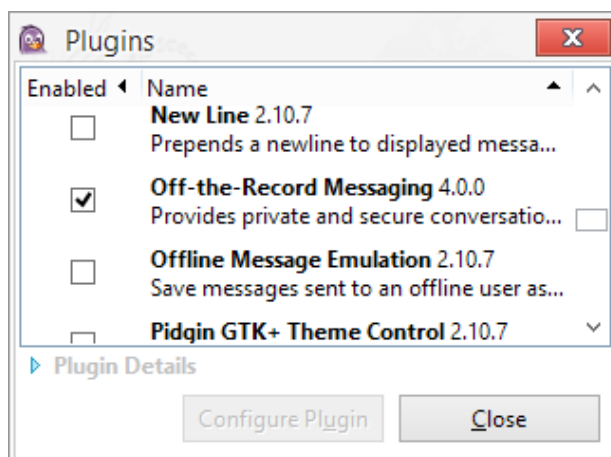
Stvořitelé OTR a OTR team stáli za vytvořením pluginu pro IM komunikátor Pidgin.

Ten byl již popsán v sekci 5.3.1 a plugin je dostupný na webu OTR teamu, v sekci download. [26] Pro otestování musel být program nainstalován. Zvoleny byly defaultní lokace a pro otestování přenosu zpráv byly zaregistrovány dva uživatelské účty na serveru jabber.cz



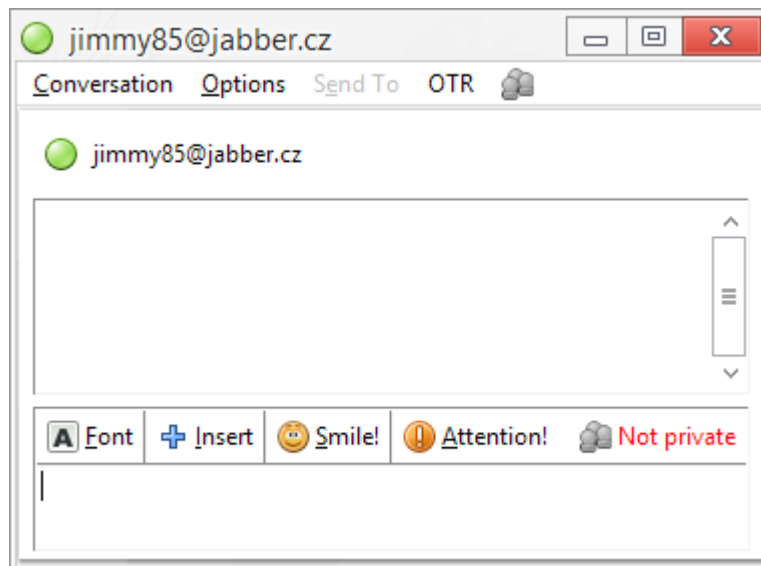
Spuštěný IM Pidgin s listem přidanych kontaktů. Obrázek 7-4

OTR plugin bylo nutné nainstalovat a to pomocí připraveného instalátoru pro OS Windows. Při instalaci bylo nutné zvolení správného kořenového adresáře použitého při instalaci samotného IM komunikátoru. Po instalaci samotné bylo nutné restartování samotného programu a následná aktivace pluginu.



Aktivace OTR pluginu. Obrázek 7-5

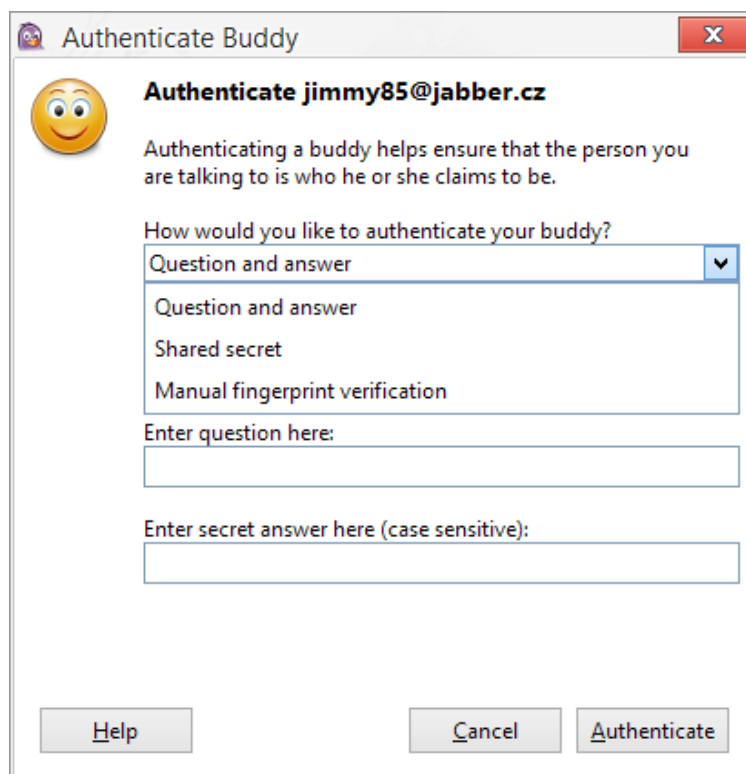
Tato aktivace stála za přidáním nové položky kontextové nabídky komunikačního okna konverzace. Ověření uživatelů zatím nebylo provedeno, proto je zobrazována informace Not private.



Okno konverzace s, zatím, neověřenou identitou uživatelů. Obrázek 7-6

Pro ověření uživatelů musí dojít ke generaci privátních klíčů (viz sekce 3.1.3). Toho je dosaženo automaticky po zvolení Authenticate Buddy ať už pomocí kontextové nabídky OTR nebo tlačítka zobrazujícího v tomto okamžiku informaci Not private.

V tomto pluginu došlo k implementaci Socialist Millionaires' protokolu a možnosti zadání vlastní otázky a správné odpovědi.



Možnost volby ověření. Obrázek 7-7

Jakmile je ověření dokončeno a dialogové okno to potvrzující je potvrzeno, je signalizováno použití OTR protokolu a tedy spuštění soukromé komunikace – pomocí zprávy Private.



Okno konverzace s již aktivovaným módem soukromé komunikace. Obrázek 7-8

ZÁVĚR

Prostředí OTR protokolu nabízí neobvyklý přístup k šifrování komunikace, pro uživatele přináší nejenom možnost šifrování vlastní zprávy, ale díky autentizaci účastníků také možnost ověření identity druhé strany a anonymitu při nemožnosti zpětného prokázání odesílatele zprávy.

Teoretická část byla věnována historii šifrování, pojmům steganografie a kryptografie, vzniku moderních šifrovacích algoritmů a to, jak a k čemu jsou tyto prvky využívány v různých úkolech. Pro prostředí OTR protokolu bylo nutné definování prostředků potřebných k realizaci a funkci OTR protokolu. Následoval vlastní popis principu první verze OTR protokolu. Vysvětlen byl také Man in the Middle Attack způsob útoku, který stál za důvody pro vytvoření OTR protokolu verze 2. a popsání rozdílů oproti první verzi.

V praktické části je na mé vlastní implementaci OTR knihovny na instant messaging aplikaci ukázán průběh ověřování sdíleného tajemství a výměna zprávy. V druhé části je popsána používaná implementace na programu Pidgin.

Integrace OTR knihovny proběhla na úrovni úprav jádra komunikátoru až při běhu samotné aplikace, vnitřní možnosti přizpůsobení běhu komunikátoru se ukázaly nevhodné a úprava skriptů funkcí samotných měla za následek nefunkčnost aplikace, pravděpodobně z důvodu minimalizace knihoven pro rychlejší načítání a nekompatibilitosti provedených úprav. Instalace a spuštění komunikátoru bylo provedeno na lokální úrovni a to kvůli potřebným nestandardním nastavením web serveru, které zvolené web hostingové programy nepodporují. Lokálně byl spuštěn i XMPP server. Následovalo by prozkoumání nabídek provozu vlastních serverů, zda takové řešení dovoluje nízkonákladové požadavky a případné spuštění projektu online.

Již implementované řešení v programu Pidgin je plně funkční a okamžitě použitelný. Nutná je ovšem instalace programu a pluginu samotného, je dostupné pouze tam, kde je aplikace nainstalována. Navrhované řešení implementace OTR knihovny na webový instant messenger Candy zaručuje uživatelům využití soukromé komunikace všude, kde je nainstalován webový prohlížeč (každý počítač s připojením k internetu). V jednoduchosti použití tedy Pidgin předčí.

ZÁVĚR V ANGLIČTINĚ

The OTR protocol environment gives unusual access to encryption communication. It brings to the users not only the encryption of the message, but thanks to user authentication, also the opportunity of identity verification of the opposite side and anonymity when the other side cannot be verified.

Theoretical part is devoted to the history of encryption, to concepts of steganography and cryptography, to development of modern encryption algorithms and how and to what they are used for different tasks. For OTR protocol environment was necessary to define needed resources for OTR protocol realization and usage. Followed by description of the first version of OTR protocol principles. Also was explained the "Man in the middle" attack, which was the reason to develop the OTR protocol version 2nd and describing differences compared to the first version.

In the practical part is shown the process of shared secret verification and message exchange on my own implementation of OTR library on „instant messaging“. In the second part is described used implementation of Pidgin application. The integration of OTR library was based on core adjustment of communicator during the its running. Inner adjustments options of the communicator proved unsuitable and adjustment of scripts itself resulted in a application failure due to minimization of libraries for faster loading. The minimization was not apparently compatible with the script adjustment. Installation and running the communicator was performed on local site because of individual nonstandard settings of web server, which chosen web hosting clients does not support. A XMPP server was also running locally. Followed by offers of own servers service, whether such solution allows low cost demands and possible running the project online.

Now implemented solution in Pidgin application is fully functional ready for use. However it is necessary to install the application and Pidgin itself. Therefore it can be used only when it is installed locally. Designed OTR library implementation on web based instant messenger Candy ensures the users it can be used everywhere where the web browser is installed (basically all computers with internet access). This is the biggest advantage of my solution and combined with its ease of use it surpasses the Pidgin application.

SEZNAM POUŽITÉ LITERATURY

- [1] Arlolra/otr · GitHub. BREault, Arlo. *GitHub · Build software better, together*. [online]. 2012 [cit. 2013-06-14]. Dostupné z: <https://github.com/arlolra/otr>
- [2] BELLARE M., R. CANETTI a H. KRAWCZYK. HMAC: *Keyed-hashing for message authentication*, RFC2104. 1997. [online]. [cit. 2012-06-11]. Dostupné z: <http://www.ietf.org/rfc/rfc2104.txt>.
- [3] *Bezplatná internetová volání Skype a levná volání na telefonní čísla online – Skype* [online]. 2013 [cit. 2013-06-14]. Dostupné z: <http://www.skype.com/cs/>
- [4] *Candy — a JavaScript-based multi-user chat client* [online]. 2012 [cit. 2013-06-14]. Dostupné z: <http://candy-chat.github.io/candy/#about>
- [5] DENIS, Tom St. a Simon JOHNSON. *Cryptography for developers*. Rockland, Mass.: Syngress, 2007, xxii, 423 s. ISBN 978-159-7491-044.
- [6] DI RAIMONDO M., R. GENNARO a H. KRAWCZYK. *Secure Off-the-Record Messaging*. In *Proceedings of the 2005 ACM Workshop on Privacy in the Electronic Society*, ACM Press, 2005, s.81-89
- [7] DIFFIE W. a M. E. HELLMAN. *New Directions in Cryptography*. In *IEEE Transactions on Information Theory*, vol. IT-22, 1976, s.644-654.
- [8] DIFFIE W., P. C. VAN OORSCHOT a M. J. WIENER. *Authentication and authenticated key exchanges*. In *Designs, Codes and Cryptography*, 1992, s.107-125.
- [9] GOLDBERD, Ian et al. *Off-the-Record Messaging* [online]. [cit. 2013-05-02]. Dostupné z: <http://www.cypherpunks.ca/otr/>
- [10] GROŠEK, Otokar. *Základy kryptografie*. 1. vyd. Bratislava: Vydavateľstvo STU, 2006, iv, 184 s. ISBN 80-227-2415-7.
- [11] HowStuffWorks "How Instant Messaging Works". TYSON, Jeff a Alison COOPER. *HowStuffWorks "Learn how Everything Works!* [online]. [cit. 2013-06-14]. Dostupné z: <http://computer.howstuffworks.com/e-mail-messaging/instant-messaging.htm>
- [12] HUGH SEBAG-MONTEFIORE. *Enigma - bitva o kód*. Brno: B4U Publishing, 2009. ISBN 978-80-87222-09-6.
- [13] ICQ 8 s levným telefonováním míří k českým uživatelům. Má se Skype bát? - Lupa.cz. MACICH, Jiří. *Lupa.cz - server o českém Internetu* [online]. 2013 [cit.

- 2013-06-14]. Dostupné z: <http://www.lupa.cz/clanky/icq-8-s-levnym-telefonovanim-miri-k-ceskym-uzivatelum-ma-se-skype-bat/?labelsBox-labelId=809&do=labelsBox-switch>
- [14] Ignite Realtime: Openfire Server. *Ignite Realtime: a real time collaboration community site* [online]. 2012 [cit. 2013-06-14]. Dostupné z: <http://www.igniterealtime.org/projects/openfire/>
- [15] *Imo messenger: about* [online]. 2012 [cit. 2013-06-14]. Dostupné z: <https://imo.im/about.html>
- [16] JANEČEK, Jiří. *Rozluštěná tajemství: luštitelé, dešifranti, kódy a odhalení*. Vyd. 2. V Praze: XYZ, 2008, 268 s. ISBN 978-80-86864-96-9.
- [17] JOHNSON, Neil F. *Steganography: Seeing the Unseen*. JOHNSON, Neil F. a Sushil JAJODIA. Johnson & Johnson Technology Consultants, LLC [online]. 1996, 1998 [cit. 2013-06-10]. Dostupné z: <http://jjtc.com/pub/r2026a.htm>
- [18] KAHN, David. *The codebreakers*. Repr. London: Sphere Books, 1977, xvi, 476 s. ISBN 0-7221-5149-7.
- [19] KATZ, Jonathan a Yehuda LINDELL. *Introduction to modern cryptography*. Boca Raton: Chapman & Hall/CRC, 2008, xviii, 534 s. ISBN 978-1-58488-551-1.
- [20] KIAYIAS, Aggelos a Serdar PEHLIVANOGLU. *Encryption for digital content*. New York: Springer, 2010, xiii, 209 s. ISBN 978-1-4419-0044-9.
- [21] KRAWCZYK, Hugo *Sigma: 'SIGn-and-Mac' Approach to Authenticated Diffie-Hellman and Its Use in IKE Protocols*, In *Proceedings of CRYPTO*, 2003, s. 400-425
- [22] Man in the Middle Attack. *Application Security Testing / Veracode* [online]. 2005 [cit. 2013-06-14]. Dostupné z: <http://www.veracode.com/security/man-in-the-middle-attack>
- [23] *Messenger for Windows* [online]. 2012 [cit. 2013-06-14]. Dostupné z: <https://www.facebook.com/about/messenger>
- [24] *Miranda IM - Home of the Miranda IM client. Smaller, Faster, Easier* [online]. 2012 [cit. 2013-06-14]. Dostupné z: <http://www.miranda-im.org/>
- [25] MOLDOVYAN, Nick a Alex MOLDOVYAN. *Innovative cryptography*. 2nd ed. Boston: Charles River Media, 2007, xiii, 386 p. ISBN 978-158-4504-672.
- [26] Off-the-Record Messaging - Software. *Off-the-Record Messaging* [online]. 2005 [cit. 2013-06-14]. Dostupné z: <http://www.cypherpunks.ca/otr/index.php#downloads>

- [27] Off-the-Record Messaging Protocol version 2. *Off-the-Record Messaging* [online]. 2005, 2012 [cit. 2013-06-14]. Dostupné z: <http://www.cypherpunks.ca/otr/Protocol-v2-3.1.0.html>
- [28] *Pidgin, the universal chat client* [online]. 2012 [cit. 2013-06-14]. Dostupné z: <http://pidgin.im/>
- [29] PIPER, F a Sean MURPHY. *Kryptografie*. 1. vyd. v českém jazyce. Praha: Dokořán, 2006, 157 s. ISBN 80-7363-074-5.
- [30] POLESNÝ, David. ICQ, Jabber, Gtalk, Skype: co je nejlepší? - 2 .kapitola – Živě.cz. *Živě.cz – O počítačích, IT a internetu* [online]. 2011 [cit. 2013-06-14]. Dostupné z: http://www.zive.cz/clanky/icq-jabber-gtalk-skype-co-je-nejlepsi/doporucene-programy-pro-instant-messaging/sc-3-a-158143-ch-77311/default.aspx#utm_medium=navigation&utm_source=zive&utm_campaign=nexchapter
- [31] QIP 2012 download - Slunečnice.cz - programy rychle a zadarmo. *Slunečnice.cz - programy rychle a zadarmo* [online]. 2013 [cit. 2013-06-14]. Dostupné z: <http://www.slunecnice.cz/sw/qip/>
- [32] RSA Laboratories - 2.1.7 What are Message Authentication Codes?. *RSA Laboratories* [online]. 2012 [cit. 2013-06-14]. Dostupné z: <http://www.rsa.com/rsalabs/node.asp?id=2177>
- [33] SINGH, Simon. *Kniha kódů a šifer: tajná komunikace od starého Egypta po kvantovou kryptografii*. Praha: Dokořán, 2003, 382 s. ISBN 80-865-6918-7.
- [34] The Secure Hash Algorithm Directory - MD5, SHA-1, HMAC and other Cryptography Resources. *The Secure Hash Algorithm Directory* [online]. 2000, 2012 [cit. 2013-06-14]. Dostupné z: <http://www.secure-hash-algorithm-md5-sha-1.co.uk/>
- [35] VÍT, Svatopluk. Instant messaging pro začátečníky aneb IM síť - Root.cz. *Root.cz - informace nejen ze světa Linuxu* [online]. 2008 [cit. 2013-06-14]. Dostupné z: <http://www.root.cz/clanky/instant-messaging-pro-zacatecniky-aneb-im-site/>
- [36] VLČEK, Karel. *Teorie informace, kódování a kryptografie*. 1. vyd. Ostrava: VŠB-Technická univerzita, 1999, 182 s. ISBN 80-7078-614-0.
- [37] VONDRUŠKA, Pavel. *Kryptologie, šifrování a tajná písma*. Praha: Albatros, 2006. ISBN 8000018888.

- [38] *Welcome! - The Apache HTTP Server Project* [online]. 2013 [cit. 2013-06-14].
Dostupné z: <http://httpd.apache.org/>
- [39] Windows Live Messenger. *Free software downloads and reviews - Softonic* [online]. 2012 [cit. 2013-06-14]. Dostupné z: <http://windows-live-messenger.en.softonic.com/>
- [40] ZELENKA, Josef. *Ochrana dat: kryptologie*. Vyd. 1. Hradec Králové: Gaudeamus, 2003, 197 s. ISBN 80-7041-737-4.

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

AES	Advanced Encryption Standard
AES-CTR	AES counter mode
AKE	Authenticated Key Exchange
CTR	Counter mode
DES	Data Encryption Standard
3DES	Triple Data Encryption Standard
DH	Diffie-Hellman
g	Generátor grupy
h	Hashovací funkce
IM	Instant Messaging
MAC	Message Authentication Code
MITM	Man-in-the-middle
NITS	Nation Institute of Standards and Technology
NSA	National Security Agency
OTR	Off-The-Record
p	Prvočíslo
PGP	Pretty Good Privacy
SHA	Secure Hash Algorithm
OS	Operating system

SEZNAM OBRÁZKŮ

Pidgin IM Obrázek 5-1	33
Prostředí IM Skype Obrázek 5-2	34
Windows Live Messenger Obrázek 5-3.....	35
Okno zprávy a seznam kontaktů IM QIP. Obrázek 5-4.....	35
IM klient ICQ. Obrázek 5-5.....	36
IM klient Facebook pro OS Windows. Obrázek 5-6	36
Miranda IM Obrázek 5-7	37
Web IM IMO. Obrázek 5-8	38
Desktopový klient GTalk. Obrázek 5-9	38
Logovací formulář pro vstup do Candy IM (metoda Nick, bez hesla). Obrázek 6-1	43
Společná místnost pro chat s listem dostupných uživatelů. Obrázek 6-2	44
Okno privátního chatu. Obrázek 6-3.....	44
Test funkčnosti OTR knihovny. Obrázek 6-4.....	45
Servisní okno Apache http serveru. Obrázek 6-5	46
Uživatelské rozhraní pro konfiguraci Openfire. Obrázek 6-6	47
Dva aktivní uživatelé v Candy IM. Obrázek 7-1	49
Inicializace OTR komunikace zasláním initotr Obrázek 7-2.....	49
Obsah inicializační zprávy Obrázek 7-3	50
Spuštěný IM Pidgin s listem přidáných kontaktů. Obrázek 7-4	53
Aktivace OTR pluginu. Obrázek 7-5	53
Okno konverzace s, zatím, neověřenou identitou uživatelů. Obrázek 7-6	54
Možnost volby ověření. Obrázek 7-7	55
Okno konverzace s již aktivovaným modelem soukromé komunikace. Obrázek 7-8.....	55

SEZNAM PŘÍLOH

P I. CD

PŘÍLOHA P I: CD

Bp-jakub-skrivanek.pdf Vypracovaná bakalářská práce

Prakticka-cast.zip Zdrojové kódy aplikace a knihovny třetích stran