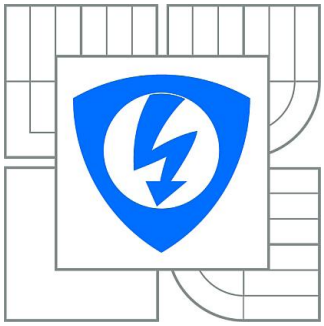
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMUNICATIONS

BEZPEČNÉ ŘÍZENÍ VZDÁLENÉ STANICE

REMOTE CLIENT SECURITY

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — Bc. JIŘÍ FRIEDBERG
AUTHOR

VEDOUCÍ PRÁCE — Ing. LUKÁŠ MALINA
SUPERVISOR

BRNO 2011

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky a komunikačních technologií

Ústav telekomunikací

Diplomová práce

magisterský navazující studijní obor
Telekomunikační a informační technika

Student:	Bc. Jiří Friedberg	***ID:***	78099
Ročník:	2	***Akademický rok:***	2010/2011

NÁZEV TÉMATU:

Bezpečné řízení vzdálené stanice

POKYNY PRO VYPRACOVÁNÍ:

Diplomová práce bude analyzovat současné metody a protokoly řízení vzdálených stanic. Student se zaměří na bezpečnostní funkce používaných technologií, protokolů a programů, dále na jejich slabiny a srovná jednotlivé technologie a programy podle jejich výhod a nevýhod. Dále provede návrh aplikace pro ovládání vzdálené stanice, která by byla funkční, bezpečná, efektivní a jednoduchá pro uživatele. Cílem diplomové práce je prakticky implementovat navržené řešení na vybrané platformě a otestovat vlastnosti zabezpečení implementace.

DOPORUČENÁ LITERATURA:

[1] Pužmanová, R.: Moderní komunikační sítě od A do Z, 2. vydání. Computer Press, Brno 2006, ISBN 80-251-1278-0.

[2] Vlček, K.: Komprese a kódová zabezpečení v multimediálních komunikacích. BEN, Praha 2004, ISBN 80-7300-134-9.

Termín zadání:	7.2.2011	***Termín odevzdání:***	26.5.2011

Vedoucí práce: Ing. Lukáš Malina

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

UPOZORNĚNÍ:

Abstrakt

Tato diplomová práce je věnována bezpečnému řízení vzdálené stanice. Bezpečnost přenosu dat a elektronických služeb je v dnešní době velmi důležitá, jelikož útoků na počítačové sítě stále přibývá. Bezpečnost lze zajistit několika různými způsoby, které lze vzájemně kombinovat. Základní obranou je autentizace uživatele a šifrování dat. Oba tyto způsoby obsahuje i vlastní implementace řízení vzdálené stanice, která je hlavní částí této práce. Uživatel výsledné aplikace má na výběr z několika typů autentizace a několika druhů šifrování. Šifrován je veškerý provoz od přenosu obrazu vzdálené plochy až po přenos souborů. Míra zabezpečení záleží pouze na uživateli, respektive na tom, jaký typ autentizace a šifrování zvolí. Práce se též zabývá analýzou implementovaných bezpečnostních funkcí. Přináší výsledky analýzy vlivu šifrovacích algoritmů na datový přenos vzdálené plochy.

Klíčová slova:

řízení vzdálené stanice, řízení vzdálené plochy, šifrování dat, autentizace, Java

Abstract

This master’s thesis is devoted to safe operating remote station. The security of data and electronic services is very important nowadays because the number of attacks to computer networks is increasing. The security of network services and information can be provided in several different ways which can be combined. The basic security mechanisms are authentication and data encryption. Both these methods are included to the implementation of operating remote station which is the main part of this work. The user of the resulting application will have a choice from several methods of authentication and data encryption. Whole traffic is encryption from the transfer of view of remote desktop to the transfer of files. Safety factor depends only up to the user consequently on that what type of authentication and encryption the user choose. The thesis also deals with analysis of implemented safely functions. Brings the results of analysis of effect of encryption algorithms to data connection of remote desktop.

Key words:

operating remote station, operating remote desktop, data encryption, authentication, Java

Bibliografická citace:

FRIEDBERG, J. *Bezpečné řízení vzdálené stanice.* Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2011. 57 s. Vedoucí diplomové práce Ing. Lukáš. Malina.

Prohlášení

Prohlašuji, že svou diplomovou práci na téma „Bezpečné řízení vzdálené stanice“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

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

V Brně dne …….. ……..

podpis autora

Poděkování

Děkuji vedoucímu práce Ing. Lukáši Malinovi za velmi užitečnou metodickou pomoc a cenné rady při zpracování diplomové práce.

V Brně dne …….. ……..
podpis autora

Obsah

Úvod

Řízení vzdálené stanice je v dnešní době moderních technologií stále využívanější technikou, která nabízí stále více možností. Dokonce i některé mobilní telefony dnes umožňují vzdáleně ovládat stanice. Toto řešení je velmi výhodné z hlediska úspory času, např. při technické podpoře různých softwarových produktů. Důležitým faktorem v této oblasti je bezpečnost. Jistě by nikdo nechtěl, aby měl každý možnost připojit se do jeho počítače a ukrást mu jeho data. Z tohoto důvodu je nutné použít nějaký způsob autentizace uživatele, aby bylo jasné, že má právo pro přístup na vzdálenou stanici. Dalším druhem ochrany je šifrování dat při přenosu komunikačním kanálem. Při použití kvalitního šifrovacího algoritmu je zjištění významu dat zachycených útočníkem téměř nemožné.

V první kapitole práce jsou nejprve popsány účely vzdáleného řízení stanic. Jsou zde uvedeny základní výhody použití i případné problémy při realizaci. V další části první kapitoly jsou popsány základní kryptografické prvky, které jsou zcela nezbytné pro pochopení celé práce. Některé, ze zde uvedených algoritmů, jsou použity při návrhu implementace a jsou použity ve výsledné aplikaci. Poslední část první kapitoly tvoří popis vybraných existujících řešení. Popis je rozdělen do tří kategorií pro lepší orientaci při případném srovnávání.

Další dvě kapitoly obsahují popis protokolů, které jsou použitelné při realizaci řízení vzdálené stanice. První protokol RFB (Remote Framebuffer) je velmi jednoduchý. Je velmi dobře specifikován a pro realizaci vzdáleného řízení je plně postačující. Druhý protokol RDP (Remote Desktop Protokol) pochází z dílny firmy Microsoft a jedná se o velmi složitý protokol.

Poslední kapitola této práce obsahuje její stěžejní část, což je popis vlastní implementace bezpečného řízení vzdálené stanice. Jsou v ní popsány veškeré možnosti nastavení aplikace. Největší pozornost je věnována popisu bezpečnostních funkcí, jejichž implementace byla úkolem této práce. Uživatel sám má možnost výběru míry zabezpečení. Jsou mu nabídnuty čtyři typy autentizace a šest druhů šifrování. Veškeré typy autentizace a druhy šifrování lze libovolně kombinovat. V další části této kapitoly je provedena analýza implementovaných druhů šifrování. Je zde uvedeno, jak se vybrané šifrování projevuje v době přenosu jednoho snímku. Dále jsou zde provedeny analýzy doby obnovy obrazu vzdálené

plochy pro praktickou použitelnost aplikace při použití jednotlivých druhů šifrování. Poslední podkapitola nabízí náměty na možná rozšíření funkcí aplikace včetně jejich stručného popisu.

1 Analýza současného řešení řízení vzdálené stanice

V první kapitole celé práce jsou vysvětleny některé základní pojmy, které se prolínají celou prací a jejich pochopení je nezbytné pro porozumění dalším kapitolám této práce. V první podkapitole jsou předkládány různé důvody pro použití řízení vzdálené stanice a jejich výhody. Další podkapitola objasňuje některé zásadní pojmy z kryptografie, která má v systémech vzdáleného řízení stanic velmi důležitou roli, protože při absenci šifrování dat by byla celá komunikace velmi lehce odposlechnutelná. Třetí podkapitola se zabývá analýzou již existujících řešení pro ovládání vzdálené plochy. Na každé existující řešení je nahlíženo hned z několika pohledů (funkce, bezpečnostní prvky, atd.). Poslední podkapitola poté přináší shrnutí analýzy současných řešení.

1.1 Účel a motivace

V části 1.1 jsou uvedeny možnosti využití řízení vzdálené stanice. Možností využití existuje mnoho a cílem kapitoly není všechny podrobně popsat, ale pouze představit jejich účel a výhody. Bezesporu největší výhodou, která je zároveň společná pro všechny níže popsané možnosti využití, je to, že smazává geografické rozdíly mezi klientem a serverem. Uživatel, který se na server připojí, od něj může být vzdálen tisíce kilometrů. K připojení uživateli stačí pouze dostatečně rychlé připojení k internetové síti.

1.1.1 Technická podpora

Technická podpora je základní a nejspíš i nejvíce využívanou možností použití vzdálené plochy. Již několik verzí je i jednou ze základních funkcí Microsoft Windows, tudíž není nutná instalace žádného klienta. Pokud nastane nějakým problém, se kterým si uživatel neví rady, stačí kontaktovat příslušného pracovníka technické podpory, povolit mu přístup do počítače a pouze čekat, než technik problém vyřeší.

Z výše uvedeného příkladu plynou hned dvě základní výhody. První výhodou je možnost vzdáleného ovládání počítače ze sídla pracovníka technické podpory, což velmi snižuje časovou náročnost řešení, i náklady na případnou dopravu. Druhou výhodou je řešení problému přímo technikem, nikoli pouze poskytování rad koncovému uživateli. To také přináší úsporu času, protože nemohou vzniknout další nečekané problémy vlivem špatného plnění pokynů od technika koncovým uživatelem.

1.1.2 Práce z domova

Další, dnes čím dál více rozšířenou, možností využití vzdálené plochy, je práce z domova. Uživatel se může připojit k pracovnímu počítači z domácího počítače a získat tak přístup ke všem souborům, programům a síťovým prostředkům, jako by pracoval přímo u počítače v práci.

1.1.3 Školení na dálku

Školení na dálku je dnes velmi se rozrůstající oblastí. Pro tento účel je ovšem velmi vhodné zajistit i kvalitní přenos zvuku, aby byla výuka interaktivní a školitel mohl přímo reagovat na veškeré dotazy. Uživatelé připojení ke vzdálené ploše školitele mohou sledovat přímo školitelovu činnost v jeho počítači a případně sami zkoušet vyučované postupy v druhém okně na svém počítači. Pokud se vyskytne nějaký problém, může se školitel případně vzdáleně připojit přímo k danému uživateli a problém sám vyřešit, zatímco uživatel sleduje jeho činnost na svém počítači. Tím se uživatel též učí.

1.1.4 Ovládání počítače přes mobilní telefon

V dnešní době rychlého vývoje elektroniky a nových technologií již není problém vzdáleně ovládat počítač i přes PDA, nebo mobilní telefon. Velká řada tzv. chytrých telefonů dnes pracuje na různých platformách (MS Windows Mobile, Android, atd.). Jejich počet se mezi uživateli neustále zvyšuje, a proto i společnosti zabývající se vývojem programů pro správu vzdálené plochy na tento trend reagují. Většina dnes nabízených programů je vyvíjena i pro platformu MS Windows Mobile a více než polovina pro Android. Z mobilního telefonu se lze k počítači připojit pomocí WIFI, EDGE, nebo GPRS. Připojení přes GPRS je ovšem velmi pomalé pro tyto aplikace. Tento způsob ovládání počítače je hojně využívaný lidmi nacházejícími se mimo domov. Když například naleznou na internetu nějaký velký objem dat, která chtějí mít, mohou se vzdáleně připojit ke svému domácímu počítači. Na něm vzdáleně spustí stahování, a jakmile se vrátí domů, data jsou již stažena.

1.2 Kryptografické prvky

Kryptografie je při využívání vzdálené plochy velmi důležitým prvkem. Chrání celou komunikaci mezi klientem a serverem proti útokům ze sítě. Kryptografických prvků existuje celá řada. V kapitole 1.2 jsou objasněny pouze základní prvky, které budou zmiňovány v dalších částech této práce a budou tvořit elementy ve výsledné implementaci.

1.2.1 Symetrické kryptosystémy

Tyto systémy používají tajný klíč. Šifrující i dešifrující klíč musí být držen oběma komunikujícími stranami v tajnosti, jelikož oba klíče jsou stejné, nebo od sebe velmi jednoduše odvoditelné. Symetrické kryptosystémy jsou velmi rychlé a používají se k zajištění důvěryhodnosti a autentičnosti zpráv. Problémem je bezpečná distribuce klíčů, viz [4].

Symetrické kryptosystémy lze rozdělit na dva základní typy: proudové šifry a blokové šifry. Rozdíl mezi oběma druhy spočívá v tom, že proudové šifry šifrují každý znak abecedy samostatně, kdežto blokové šifry šifrují najednou bloky znaků o určité délce. Podstatou blokových šifer je to, že šifrování i dešifrování probíhá pomocí stejné transformace E_k (D_k), kde k je šifrovací klíč. Oproti tomu proudové šifry nejprve z klíče k vygenerují posloupnost h(1), h(2), atd. Každý znak textu je poté šifrován jinou transformací $E_{h(i)}$, podrobněji v [1].

1.2.2 Asymetrické kryptosystémy

Systémy s veřejným klíčem. Veřejný a soukromý klíč jsou odlišné a od sebe navzájem prakticky neodvoditelné. Jeden je používán na dešifrování a druhý na šifrování podle situace. Skutečnost, který z klíčů je dešifrovací určuje, zda je zajišťována důvěryhodnost, nebo autentičnost. Důvěryhodnost je zajišťována v případě, pokud je zpráva zašifrována veřejným klíčem. Dešifrovat ji může pouze majitel soukromého klíče. Naopak autentičnost (tzv. digitální podpis) je zajištěna v případě, že je soukromý klíč použit k zašifrování zprávy, kterou může dešifrovat kdokoli pomocí veřejného klíče. Výhodou asymetrických systémů je jednodušší distribuce klíčů. Soukromý si uchová majitel v tajnosti a veřejný jednoduše zveřejní. Nevýhodou ovšem je pomalý proces šifrování a dešifrování. Asymetrické systémy se používají k podepisování zpráv a k šifrování klíčů symetrických systémů. Klíče symetrických systémů jsou velmi krátké (128 bitů), a proto není pomalé šifrování a dešifrování kritickým problémem, viz [4].

1.2.3 Data Encryption Standard (DES)

Standard DES je založen na symetrických blokových šifrách, tedy používá pro šifrování i dešifrování stejný klíč. Data šifruje v 64-bitových blocích pomocí 56-bitového klíče. Klíč se obvykle vyjadřuje jako 64-bitové číslo, každý osmý bit je ale ignorován. Ignorované bity jsou použity jako paritní zabezpečení. DES je založen na síle klíče. Z tohoto důvodu se klíče považované za slabé z výběru klíče vylučují. Celý algoritmus je založen na opakování

šestnácti šifrovacích kroků, v nichž je za použití klíče použita substituce s následnou permutací. Tento krok se v celém algoritmu opakuje šestnáctkrát, viz [2].

Vzhledem k délce klíče, která je pouhých 56 bitů, dnes není algoritmus DES považován za bezpečný. Je dokonce hrubou silou prolomitelný za dobu kratší než 24 hodin. Hranice bezpečnosti se v dnešní době udává délkou klíče minimálně 80 bitů. Proto jsou dnes používány bezpečnější verze standardu DES, například tzv. triple DES (TDES, 3DES). Jedná se prakticky o trojnásobnou aplikaci DES a délka klíče se tím zvýší na 168 bitů. Tato šifra se používá například při šifrování SSL (Secure Socket Layer) spojení, viz [5].

1.2.4 Advanced Encryption Standard (AES)

AES je schválený standard, který byl udělen symetrické blokové šifře Rijndael. Jedná se o nástupce dnes již zastaralého šifrovacího standardu DES. Šifrována jsou data v blocích o pevné velikosti 128 bitů. Délka klíče dosahuje hodnot 128, 192 nebo 256 bitů. Šifra se vyznačuje velmi vysokou rychlostí šifrování, viz [1].

1.2.5 Proudová šifra RC-4

RC-4 je velmi často využívaný algoritmus, proudová šifra. Používá jej například protokol SSL k šifrování síťové komunikace nebo WEP (Wired Equivalent Privacy) k zabezpečení bezdrátových sítí. Hlavními přednostmi algoritmu RC-4 jsou jednoduchost, rychlost, snadná a efektivní implementace. RC-4 generuje pseudonáhodný proud bitů. K šifrování používá spojení náhodných bitů spolu s čistým textem. Dešifrování probíhá inverzně. Ke generování pseudonáhodného toku bitů je použit speciální postup složený ze dvou kroků:

- permutace všech 256 možností bajtů,
- dva osmibitové index-ukazatele.

Permutace je vytvářena z proměnlivé délky klíče z rozmezí 40 – 256 bitů. Následně je použita pro generaci pseudonáhodného čísla. Více podrobností o algoritmu RC-4 je možné získat z [2].

1.2.6 Bloková šifra RC-6

RC-6 je na rozdíl od RC-4 bloková šifra. Velikost bloků při použití RC-6 je 128 bitů. Podporuje klíče o velikosti 128, 196 a 256 bitů. RC-6 má širokou škálu volitelných parametrů: *w* – počet bitů slova,

r – počet rund,

b – počet bajtů klíče.

Proto se podle nich přesně označuje jako RC6-*w*/*r*/*b*. (Pro AES bylo stanoveno *w* = 32, *r* = 20, *b* = 16, 24 nebo 32 bajtů).

Vychází ze starší verze RC-5 (2 vhodně propojené šifry RC-5). Pro deklarovaný počet 20 rund je považována za bezpečnou, viz [20].

1.2.7 Hašovací funkce MD5

MD5 (Message-Digest algorithm 5) je v kryptografii velmi používaná hašovací funkce se 128-bitovou hašovací hodnotou. Je specifikována v RFC 1321 (Request For Comments). MD5 má mnoho variant použití v zabezpečování a používá se též k ověřování neporušenosti souborů. Není ovšem zcela odolný vůči kolizím, proto není vhodný k použití v SSL certifikátech a digitálních podpisech. Obvykle se vyjadřuje jako 32-místné hexadecimální číslo.

Algoritmus pracuje s daty, která mají délku v bitech rovnu násobku 512. Pokud není velikost násobkem 512, je nutné ji na tuto velikost doplnit. Doplnění se provede jedním bitem hodnoty 1 tak, aby výsledná zpráva byla o 64 bitů kratší, než je požadovaný násobek 512 bitů. Zbývajících 64 bitů slouží k uchování délky zprávy před doplněním. Doplnění se provádí i u zpráv, které mají délku v násobku 512 bitů. Po úpravě lze zjistit hodnotu „value“, která závisí na opakované modifikaci 128-bitové hodnoty popisující stav. Po zpracování je každý 128-bitový blok rozdělen na 32-bitové bloky. Na počátku se každému bloku nastaví výchozí hodnota. Každý blok je poté zpracováván nezávisle na ostatních a různě modifikován pomocí různých logických operací. Bližší popis algoritmu MD5 je k nalezení v [11].

1.2.8 Hašovací funkce SHA

SHA (Secure Hash Algorithm) je hašovací funkce, která vytváří ze vstupních dat výstup (otisk) fixní délky. Hlavní vlastností otisku je, že malá změna na vstupu vede k vytvoření zásadně odlišného otisku na výstupu.

SHA je rodina pěti algoritmů: SHA-1, SHA-224, SHA-256, SHA-384 a SHA-512. Poslední čtyři varianty jsou souhrnně označovány jako SHA-2. SHA-1 vytváří obraz délky 160 bitů, u ostatních algoritmů je délka otisku v bitech dána číslem v jejich názvu. SHA nalézá použití u mnoha protokolů a aplikací (TLS, SSL, PGP, SSH, S/MIME, IPsec). Používá se zejména pro kontrolu integrity souborů či ukládání hesel ve formě hašů a je také nasazena v TLS/SSL (Secure Socket Layer / Transport Layer Security) a jiných protokolech. V dnešní době již zřejmě vytlačila hašovací funkci MD5 (viz 1.2.7), viz [15].

SHA-1

SHA-1 je v současnosti nejvíce používaná hašovací funkce. Vytváří 160-bitový obraz zprávy s maximální délkou $2^{64} - 1$ bitů. Je založený na totožných principech jako MD5. SHA-1 je vlastně druhá hašovací funkce SHA. Původní (označovaná SHA-0) byla v krátkém čase po svém vydání stažena a opravena. Opravená verze je SHA-1. SHA-0 a SHA-1 se od sebe liší pouze jednou bitovou rotací. SHA-1 se zdá být více obranyschopná proti útokům, viz [15].

SHA-2

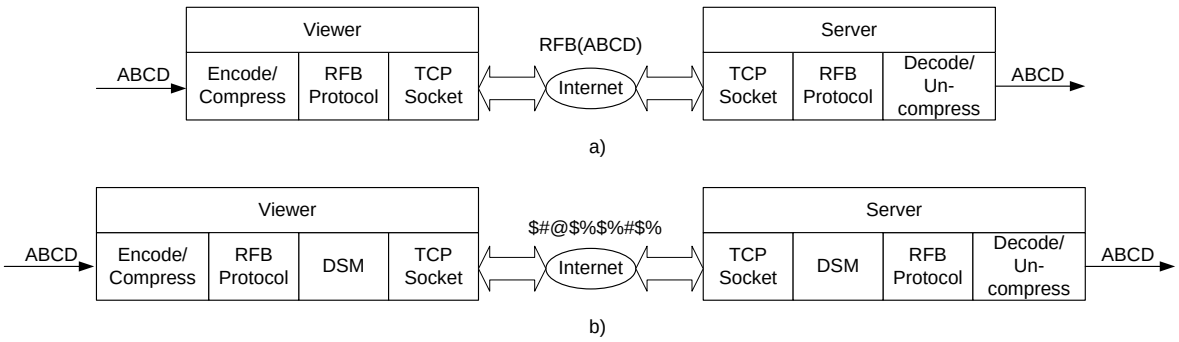
SHA-2 označuje čtyři další hašovací funkce (SHA-224, SHA-256, SHA-384 a SHA-512). Vznikla na popud kryptografických odborníků, kteří zpochybnili bezpečnost verze SHA-1. Varianta SHA-2 není doposud hojně využívána, což může být způsobeno nedostatkem podpory, nedostatečným vnímáním naléhavosti nutnosti přechodu na SHA-2 (kolize SHA-1 nebyly nalezeny), nebo čekáním na standardizaci SHA-3.

SHA-256 je používána k ověřování softwarových balíků linuxové distribuce Debian. Spolu s SHA-512 jsou navrženy pro použití v DNSSEC (Domain Name System Security Extensions), viz [15].

1.2.9 DSM plugin

DSM (Data Stream Modification) plugin byl navržen pro zrychlení, zefektivnění a menší zatížení procesoru při transformaci dat. Využívá různé externí knihovny, které slouží k úpravě a šifrování datových paketů během přenosu.

Při pouhém zapouzdření vstupní posloupnosti do protokolu RFB (Remote Framebuffer) je posloupnost při přenosu přes síť stále čitelná. Při použití DSM jsou vstupní posloupnost i obal protokolu RFB transformovány na plugin a při přenosu přes síť je přečíst nelze. Zpět jsou transformovány až na serveru. Tento rozdíl ukazuje obrázek 1.1. Více podrobností o DSM pluginu je možné nalézt v [7].



Obr. 1.1: Rozdíl v přenosu posloupnosti ABCD – a) zapouzdření do RFB, b) využití DSM

MSRC4 plugin

MSRC4 plugin je jedním ze skupiny DSM pluginů. Používá Microsoft Crypto API (Application Programming Interface) pro šifrování dat pomocí proudové šifry RC-4. Jedná se o stejnou šifrovací metodu, jaká je používána u SSL. Jako šifrovací klíč je použito 128-bitové hash heslo. Heslo je vytvořeno pomocí algoritmu MD5, viz [7].

1.3 Existující řešení řízení vzdálené stanice

V podkapitole 1.3 budou zmíněna některá již existující řešení pro použití vzdálené plochy. Existujících řešení je celá řada. Zde bude popsána jen velmi malá část z nich, která je ovšem v praxi nejvíce využívána. Popis jednotlivých řešení je rozdělen do několika kategorií, kterými jsou platforma (na které je lze spustit), nabízené funkce a bezpečnostní funkce.

Všechna níže popsaná řešení se vyznačují několika společnými vlastnostmi. Podporují více relací používaných současně na jednom serveru, také jim nečiní potíže práce za

firewallem, přes který projdou, aniž by bylo potřeba jakkoli upravovat jeho nastavení. Ani pro jedno zde uvedené řešení též není problémem vzdálená práce se stanicí, která je schována za NATem (Network Address Translation).

1.3.1 Aplikace TeamViewer

Jedná se pravděpodobně o nejpoužívanější software pro práci se vzdálenou plochou. TeamViewer využívá proprietální protokol. Pro nekomerční využití je v plné verzi zcela zdarma. Nabízí též zdaleka nejširší škálu funkcí ze zde popisovaných programů. O aplikaci TeamViewer lze nalézt bližší informace v [8].

Platforma

MS Windows, Mac OS, Linux, iPhone, MS Windows Mobile, Android

Funkce

TeamViewer nabízí celou řadu funkcí. Novinkou v dosud nejnovější verzi 5 je Audio/Video konference přes VoIP (Voice over Internet Protocol). Je zajištěna optimální kvalita obrazu a zvuku i při současném přenosu souborů, nebo aktivní práci na vzdálené ploše. Dále je zajištěna zpětná kompatibilita se staršími verzemi 3 a 4. Dopředná kompatibilita ze starších verzí na verzi 5 ale neexistuje. TeamViewer je též schopen optimalizovat výkon v závislosti na rychlosti připojení. Dále jsou nabízeny tyto funkce: týmová práce a schůzky na vzdálené ploše, prodej a prezentace z okna prohlížeče, přenos souborů.

Bezpečnostní funkce

TeamViewer pracuje s kompletním RSA šifrováním založeným na výměně šifrovacích klíčů pro symetrické algoritmy pomocí zašifrování veřejným klíčem. Výměna šifrovacích klíčů probíhá přes TeamViewer Master server a je zabezpečena 1024-bitovým RSA. Relace mezi dvěma koncovými uživateli je poté šifrována pomocí 256-bitového AES. Tato technologie je použita například v protokolu https/SSL a je v dnešní době považována za zcela bezpečnou. Výměna klíčů též zabezpečuje plnou klientskou ochranu údajů. Ani směrovací servery nejsou schopny data přečíst.

Dalším prvkem ochrany je zabezpečení přístupu. Kromě automaticky vytvářené dynamické identifikace „PartnerID“ je vytvářeno ještě heslo relace, které je při každém spuštění jiné. Jedná se o zabezpečení proti neoprávněnému přístupu do systému.

Další funkcí související se zabezpečením je zamezení neviditelnosti při práci. Z důvodu ochrany údajů ve vzdáleném počítači musí být uživatel vzdáleného počítače informován o pokusu o přístup. Též při různých funkcích (např. přenos souborů) je vyžadováno manuální potvrzení od vzdáleného uživatele.

1.3.2 Aplikace LogMeIn

Tento program opět využívá proprietální protokol. Program existuje v několika verzích. Jednou z nich je verze Free, která má ovšem omezené funkce, proto zde bude popisována verze Pro, kterou je možno zakoupit za 69,95$ ročně. Ani verze Pro ovšem nenabízí zdaleka tolik funkcí jako dříve popsaný TeamViewer, který je navíc pro nekomerční využití zdarma. O aplikaci LogMeIn lze nalézt bližší informace v [11].

Platforma

MS Windows, Mac OS, MS Windows Mobile, iPhone, Android

Funkce

LogMeIn nabízí standardní funkce jako mnoho dalších programů pro správu vzdálené plochy. Nabízí plnou podporu vzdáleného přístupu včetně přesunu souborů mezi počítači, tisku souborů ze vzdáleného počítače na lokální tiskárně, sdílení souborů a plochy s více uživateli. Podporuje i kopírování pomocí schránky.

Bezpečnostní funkce

LogMeIn využívá SSL/TLS komunikační protokol. Tento protokol poskytuje ochranu proti odposlechu, manipulaci a padělání zpráv. LogMeIn hostitelé udržují trvalé spojení s LogMeIn serverem. Totožnost LogMeIn serveru je ověřována pomocí PKI (Public Key Infrastructure) certifikátu. Totožnost hostitele je ověřena na základě předem přiřazeného identifikačního kódu. Ověřování totožnosti zabraňuje tzv. „Man in the Middle[1]“ útokům. Po řádném ověření je sestaveno spojení. Pokud certifikát nebyl vydán certifikující autoritou, nebo pokud hostitelské jméno v URL (Uniform Resource Locator) nesouhlasí s hostitelským jménem uvedeným v certifikátu, spojení není sestaveno a objeví se varovné hlášení.

[1] Man in the Middle (člověk uprostřed) je jeden z nejpoužívanějších útoků na důvěryhodnost spojení. Jeho podstatou je snaha útočníka odposlouchávat komunikaci mezi účastníky tak, že se stane aktivním prostředníkem. Důležitým faktem je, že v prostředí současných běžných počítačových sítí není nutné, aby komunikace přes útočníka přímou cestou procházela, jelikož lze síťový provoz velmi snadno přesměrovat, viz [12].

Data jsou přenášena pomocí komunikačního protokolu SSL/TLS, který nabízí velké množství šifer – RC-4 (128b), 3DES (168b) a nejčastěji používanou AES (128b nebo 256b). Klient zašle serveru list se šiframi, které mohou být použity. Server z nich vybere nejsilnější variantu a ta je poté pro přenos použita, pokud uživatel šifrování požaduje.

1.3.3 Aplikace GoToMyPC

Tento program, stejně jako oba předešlé, používá proprietální protokol. GoToMyPC nabízí k vyzkoušení třicetidenní Trial verzi, která obsahuje všechny funkce bez omezení. Plná verze programu je poté k dispozici za 7,99€ měsíčně. O aplikaci GoToMyPC lze nalézt bližší informace v [7].

Platforma

MS Windows, Mac OS, MS Windows Mobile, Android

Funkce

Opět nabízí standardní funkce, jako jsou plnohodnotný vzdálený přístup. Podporuje kopírování a vkládání textu pomocí schránky, tisk ze vzdáleného počítače, přenos souborů, zobrazování na více monitorech (tzv. multi-monitor). Také umožňuje poslech zvuků umístěných na vzdáleném počítači. Nabízí ovšem i jednu funkci, kterou předešlé programy nenabízely, a tou je možnost sledovat předešlé aktivity účtu, jako je čas, typ a průběh připojení.

Bezpečnostní funkce

Veškerá komunikace mezi koncovým klientem a serverem je šifrována pomocí 128-bitového standardu AES. Autentizace ověřuje identitu všech spojení z GoToMyPC brokeru a komunikačního serveru ke koncovému klientu a serveru. Řízení přístupu dále zabezpečuje, že do zabezpečených částí získá přístup pouze ověřený zdroj. Je zde též nastaven limitní počet pokusů o přihlášení. V základu je nastaveno, že po třech neúspěšných pokusech o autentizaci, je uživatelskému účtu a PC deaktivována možnost přístupu po dobu pěti minut. Uživatel je též automaticky odpojen z webu GoToMyPC, jestliže SSL připojení je nečinné po dobu 15 minut.

1.3.4 Aplikace UltraVNC

UltraVNC je jediný zde popsaný program, který nepoužívá proprietální protokol, nýbrž používá protokol RFB (VNC). Program je v plné verzi zdarma. O aplikaci UltraVNC lze nalézt bližší informace v [2].

Platforma

MS Windows, přes webový prohlížeč lze použít i Linux, Max OS nebo Android, pokud podporují JAVU

Funkce

UltraVNC nepodporuje velké množství funkcí. Základními podporovanými funkcemi jsou plné využití vzdálené plochy, přenos souborů a podpora zobrazování na více monitorech (tzv. multi-monitor). Z pokročilejších funkcí UltraVNC nabízí automatické obnovení spojení a textový chat mezi vzdálenými počítači.

Bezpečnostní funkce

UltraVNC má k dispozici tři druhy autentizace:

- klasická VNC – uživatelské jméno a heslo,
- MS-Logon I – uživatel se může přihlásit pouze ke vzdálenému počítači ve stejné doméně, v jaké se sám nachází,
- MS-Logon II – podporuje připojení ke vzdálenému počítači, který se nachází v jiné doméně, než uživatel, který se hodlá připojit.

Pro šifrování je použit DSM plugin, konkrétně plugin MSRC4, popřípadě některé Open SSL pluginy. Pluginy DSM a MSRC4 jsou blíže popsány v [7].

1.4 Shrnutí

Cílem první kapitoly je objasnit základní principy, účely a pojmy používané v systémech řízení vzdálených stanic.

V podkapitole Účel a motivace jsou popsány některé ze základních využití systémů vzdálených stanic, včetně jejich principů a výhod. Využití existuje celá řada, zde je uváděno ve stručném popisu pouze několik z nich.

Podkapitola Kryptografické prvky objasňuje některé základní pojmy z kryptografie, která je v oblasti vzdáleného řízení stanic velmi důležitá, jelikož útoků na všechny druhy síťových útoků stále přibývá. Jsou v ní vysvětleny některé systémy, šifry a standardy, které jsou používány v dalším textu.

V podkapitole Existující řešení řízení vzdálené stanice jsou popsány některé z mnoha již existujících řešení pro řízení vzdálené stanice. Popis řešení je pro přehlednost rozdělen do několika kategorií. Z řešení popsaných v podkapitole Existující řešení je bezesporu nejlepší TeamViewer. Jeho hlavní výhodou je využití pro nekomerční účely v plné verzi zcela zdarma. Též nabízí zdaleka nejvíce funkcí. Ostatní zde popsané programy nabízí většinou pouze základní funkce, kdežto TeamViewer nabízí například i Audio/Video konferenci pomocí VoIP. Také z hlediska bezpečnosti nabízí nejlepší parametry. Výměna klíčů probíhá přes TeamViewer Master server a je zabezpečena 1024-bitovým RSA. Relace mezi klientem a serverem je poté šifrována 256-bitovým AES. Takto zabezpečený přenos klíčů žádný jiný ze zde popisovaných programů nenabízí.

2 Protokol RFB

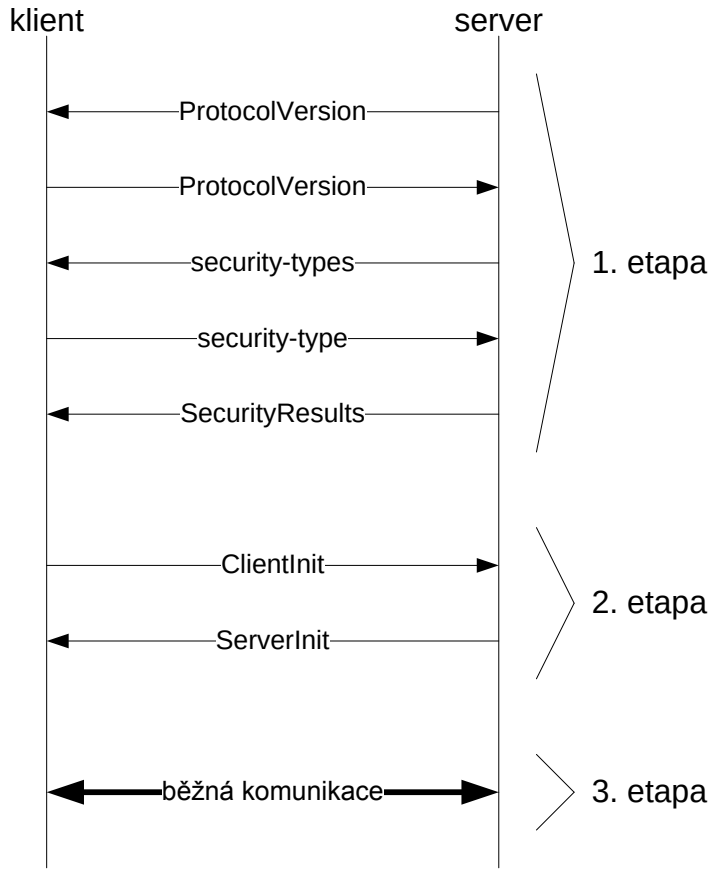
Ve druhé kapitole je podrobně popsán protokol RFB (Remote Framebuffer). Vzhledem k tomu, že je založen na framebufferu[2], je možné jej použít na nejrůznějších systémech pracujících v grafickém režimu a používajících okna (např. Microsoft Windows, MAC OS). Tento protokol je hojně využíván pro VNC (Virtual Network Computing). Pro přenos zpráv využívá spolehlivé spojení. Při použití nejrozšířenější protokolové sady TCP/IP (Transmission Control Protocol/Internet Protocol) je využit transportní protokol TCP.

Protokol je založen na komunikaci typu klient – server. Klient je počítač, u kterého jsme fyzicky přítomni, vzdálený počítač reprezentuje server. RFB protokol je navržen tak, že se stav probíhající relace ukládá na straně serveru. Pokud se tedy uživatel odpojí a poté opět připojí z jiného klienta, nalezne vzdálenou plochu v takovém stavu, v jakém ji předtím zanechal.

Navazování spojení probíhá tak, že server naslouchá na portu číslo 5900, který pro tento účel vyhradila organizace IANA (Internet Assigned Numbers Authority). Klient se přes tento port připojí k serveru, a poté proběhne výměna zpráv. Výměna zpráv je rozděleny do tří etap. První etapou je tzv. „handshaking“, při kterém je dohodnuto, jaká verze protokolu a jaký typ ochrany bude použit. Druhou etapou je inicializace, při které si klient a server vymění zprávy ClientInit a ServerInit. Třetí etapou je běžná protokolová komunikace.

Komunikace mezi klientem a serverem je znázorněna na obrázku 2.1 a popis všech zpráv je učiněn v následujících podkapitolách.

[2] Framebuffer je obrazové výstupní zařízení, které přenáší obrazovou informaci z vyrovnávací paměti (bufferu) a obsahuje právě jeden kompletní rámec dat. Informace v tomto bufferu se obvykle skládá z číselných hodnot barev pro každý pixel (obrazový bod, který se zobrazuje) na obrazovce. Barevné hodnoty jsou obvykle ukládány jednobitově (monochromatický režim), čtyřbitově s paletou, osmibitově s barevnou paletou, 16-bitově (highcolor) a 24-bitově (truecolor). Občas bývá ještě využíván alfa kanál pro uchování informace o průhlednosti pixelu. Celková velikost paměti potřebné pro práci framebufferu je závislá na barevné hloubce, velikosti palety a rozlišení výstupního signálu, viz [1].



Obr. 2.1: Tři etapy navazování spojení mezi klientem a serverem

2.1 První etapa - navazování spojení

V první etapě navazování spojení se nejprve klient se serverem dohodnou na verzi protokolu RFB, která bude pro přenos použita. Druhým krokem, který do první etapy patří, je zvolení zabezpečení, a to buď s autentizací, nebo bez ní. Podrobný popis zpráv vyměňovaných mezi serverem a klientem v obou částech této etapy následuje v dalších podkapitolách.

2.1.1 Zpráva ProtocolVersion

Existují tři používané verze RFB protokolu – 3.3, 3.7, 3.8. V této práci bude popisována nejnovější verze 3.8.

Komunikace začíná zasláním ProtocolVersion zprávy serveru klientovi. V ní je uvedeno, jakou nejvyšší verzi protokolu server podporuje. Klient odpovídá stejnou zprávou, s verzí protokolu, která bude pro komunikaci použita. Nesmí nikdy odpovědět vyšší verzí protokolu, než byla nabídnuta serverem.

Zpráva ProtocolVersion se skládá z 12 bajtů a má tvar „RFB *xxx*.*yyy*\n“, kde *xxx* je majoritní číslo verze protokol a *yyy* minoritní. Obě čísla jsou doplněna nulami. Zde popisovaná verze 3.8 má ProtocolVersion zprávu ve tvaru „RFB 003.008\n“, viz [22].

2.1.2 Zprávy Security-types a security-type

Dalším krokem je dohodnutí zabezpečení během komunikace. Nejprve je serverem odeslána zpráva security-types, ve které jsou uvedeny tři možné typy zabezpečení. Klient odešle zpět serveru zprávu o velikosti jeden bajt, kde je uvedeno, jaký typ zabezpečení bude použit. Typy zabezpečení jsou uvedeny v tabulce 2.1.

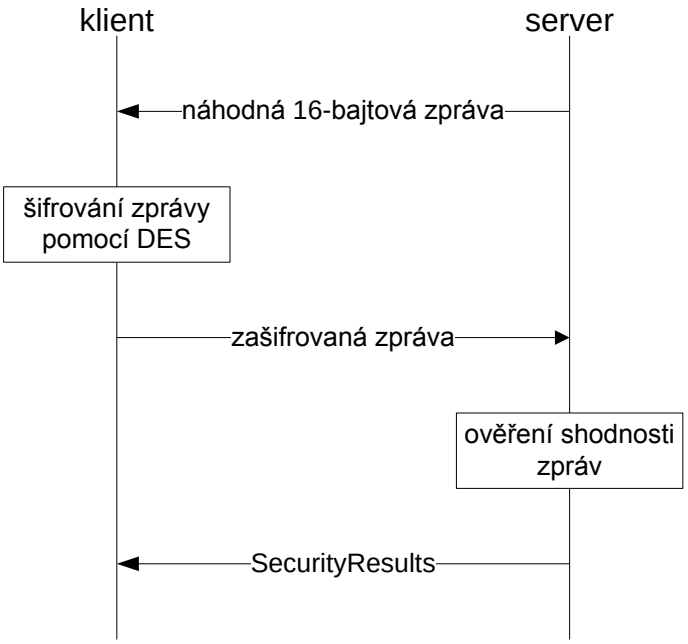
Tab. 2.1: Typy zabezpečení

číslo	název
0	Invalid
1	None
2	VNC Authentication

Číslo nula znamená, že navazování spojení bylo ukončeno. Ukončení je následováno řetězcem znaků popisujícím důvod ukončení relace.

Číslo jedna znamená, že není požadována autentizace uživatele a data budou zasílána nezašifrovaně. Protokol pokračuje komunikaci zprávou SecurityResult.

Číslo dva znamená, že autentizace uživatele je požadována, ale data budou opět zasílána nezašifrovaně. K autentizaci je použita náhodná šestnácti bajtová výzva zaslaná serverem klientovi. Klient tuto výzvu zašifruje pomocí algoritmu DES (Data Encryption Standard), jehož klíč je zadán uživatelem v podobě hesla a odešle zpět serveru jako odpověď. Server výzvu dešifruje a ověří shodnost s výzvou, kterou zaslal. Tím je autentizace ukončena a protokol pokračuje zprávou SecurityResults. Tento postup znázorňuje obrázek 2.2.



Obr. 2.2: Postup autentizace mezi klientem a serverem

2.1.3 Zpráva SecurityResults

Zprávou SecurityResults o velikosti 4 bajty informuje server, zda byl typ zabezpečení úspěšně dohodnut, či nikoli. Pokud ano, protokol povolí přechod do inicializační fáze. Pokud bylo vyjednávání neúspěšné, odešle server textový řetězec s důvodem selhání a spojení ukončí.

2.2. Druhá etapa - inicializace

Druhá etapa následuje po úspěšném ukončení etapy první. Je reprezentována dvěma zprávami, jejichž popis následuje. V nich je dohodnuto, zda se bude jednat o výlučný způsob přístupu ke stanici a parametry zobrazované plochy.

2.2.1 Zpráva ClientInit

Jednobajtová zpráva ClientInit zasílaná klientem serveru obsahuje jednu z hodnot nula nebo jedna. Pokud obsahuje číslo nula, znamená to pro server, že má odpojit všechny klienty a umožnit tomuto klientovi výlučný přístup ke stanici. V případě, že obsahuje číslo jedna, může být počítač sdílen mezi více klienty.

2.2.2 Zpráva ServerInit

Po obdržení ClientInit zprávy vysílá server zprávu ServerInit. Ta sděluje klientovi různé parametry zobrazení. Parametry zprávy jsou uvedeny v tabulce 2.2.

Tab. 2.2: Parametry zprávy ServerInit

počet bajtů	popis
2	šířka plochy
2	výška plochy
16	formát pixelů
4	délka jména
name-lenght	řetězec se jménem

Z tabulky 2.2 vyplývá, že tato zpráva má proměnnou délku, která je způsobena různou délkou jména. Šestnácti bajtové pole formát pixelů lze dále rozdělit na menší 1 - 3 bajtové části, které udávají nastavení barev, prokládání, atd. Popis všech těchto parametrů je nad rámec tohoto textu, více podrobností se můžete dozvědět v [22].

2.3 Třetí etapa – běžná komunikace

Před třetí etapou jsou mezi klientem a serverem dohodnuty všechny základní parametry komunikace. Třetí etapa slouží převážně již k běžné komunikaci, popřípadě jsou zde některé dříve dohodnuté parametry pozměněny, nebo doplněny podrobnějším nastavením.

2.3.1 Zprávy od klienta k serveru

Zprávy, které posílá klient směrem k serveru, jsou zde popsány pouze zevrubně, bližší popis je k nalezení v [22].

Tab. 2.3: Seznam zpráv zasílaných od klienta k serveru

číslo	název
0	SetPixelFormat
2	SetEncodings
3	FramebufferUpdateRequest
4	KeyEvent
5	PointerEvent
6	ClientCutText

- **SetPixelFormat** – nastavuje formát pixelů posílaných ve zprávě FramebufferUpdate. Jestliže klient neodešle zprávu SetPixelFormat, server posílá pixely v podobě, jakou uvedl ve zprávě ServerInit.
- **SetEncodings** – nastavuje kódovací typ, kterým budou data ze serveru posílána. Pokud není v této zprávě uvedeno jinak, budou data posílána v „surovém“ (základním) kódování.
- **FramebufferUpdateRequest** – jedná se o klientovu žádost o nový snímek, kde je specifikována pozice *x* a *y*, výška a šířka.
- **KeyEvent** – zpráva informuje server o stisku nebo uvolnění klávesy klientem.
- **PointerEvent** – zpráva informuje o pohybu myši, případně o stisku některého z tlačítek myši.
- **ClientCutText** – slouží pro přenos textu mezi klientem a serverem pomocí schránky. Podporováno je kódování ISO 8859-1.

2.3.2 Zprávy od serveru ke klientovi

Jednotlivé zprávy zde budou popsány opět pouze zevrubně, podrobnější popis jednotlivých zpráv je k nalezení v [22].

Tab. 2.4: Seznam zpráv zasílaných od serveru ke klientovi

číslo	název
0	FramebufferUpdate
1	SetColorMapEntries
2	Bell
3	ServerCutText

- **FramebufferUpdate** – jedná se o odpověď na žádost FramebufferRequest. Posílá se požadovaný snímek ve formátu, který byl určen klientem.
- **SetColorMapEntries** – při použití barevné mapy informuje klienta o tom, jak budou jednotlivé hodnoty pixelů namapovány do RGB intenzit.
- **Bell** – vytváří chybový zvuk.
- **ServerCutText** - slouží pro přenos textu mezi klientem a serverem pomocí schránky. Podporováno je kódování ISO 8859-1.

2.4 Kódování

Kódování je velmi důležité, protože umožňuje značně snížit objem přenášených dat. Díky tomu se výrazně snižují nároky na šířku přenosového pásma mezi klientem a serverem. Protokol RFB ve verzi 3.8 podporuje 5 základních druhů kódování a dva základní druhy tzv. pseudo-kódování (Cursor pseudo-encoding a DesktopSize pseudo-encoding). Jednotlivé druhy kódování budou blíže popsány v následujících podkapitolách. Pseudo-kódování slouží, jak již jejich název napovídá, pro přenos tvaru a velikosti kurzoru a změnu velikosti plochy.

Tab. 2.5: Přehled základních druhů kódování a pseudo-kódování

číslo	název
0	Raw
1	CopyRect
2	RRE
5	Hextile
16	ZRLE
-239	Cursor pseudo-encoding
-223	DesktopSize pseudo-encoding

2.4.1 Raw kódování

Tento typ kódování je nejjednodušší a musí být podporován všemi RFB klienty i servery. Uvede pouze výšku a šířku posílaného obdélníku a následně posílá „surová“ (raw) data, což jsou hodnoty jednotlivých pixelů.

2.4.2 CopyRect kódování

Jedná se opět o velmi jednoduché kódování, které dosahuje velké efektivity v případě, že má klient k dispozici předchozí snímek. Není totiž nutné přenášet nový snímek celý, ale stejné oblasti mohou být zkopírovány z předešlého snímku. Tento typ kódování je vhodný především na snímky s velkými plochami stejné barvy a je využíván například při scrollování.

2.4.3 RRE kódování

RRE je zkratka pro Rise-and-Run length Encoding. Jedná se opět o poměrně jednoduché kódování, které ovšem není vhodné pro složité snímky s různorodým pozadím. Přenášena je barva pozadí obdélníku, souřadnice x a y, které udávají relativní umístění přenášeného obdélníku vzhledem k levému hornímu rohu aktuálního obdélníku, a výšku a šířku obdélníku.

2.4.4 Hextile kódování

Hextile kódování vychází z RRE. Obdélníky jsou rozděleny do oblastí velikosti 16x16 pixelů. Velikost nové oblasti je pak vyjádřena pouze 4 bity. Dělení začíná v levém horním rohu a postupuje zleva doprava a shora dolů. Oblasti mohou být zakódovány dvěma způsoby, buď „surovými“ daty, nebo podobně jako v případě RRE, kdy je využita barva pozadí a barva oblasti. Všechny oblasti jsou uvozeny maskou, která udává, jakým způsobem se budou kódovat následující data. Možnosti jsou následující:

- **Raw** – následují „surová“ data. Počet pixelů udává součin šířky a výšky oblasti.
- **BackgroundSpecified** – udává barvu pozadí, celé oblasti. Pokud není hodnota nastavena, barva pozadí zůstane stejná jako v předešlé oblasti.
- **ForegroungSpecified** – zde je nastavena barva popředí, která má být použita na všechny podoblasti. Pokud je zde nastavena nějaká hodnota, SubrectsColoured musí být nula.
- **AnySubrects** – pokud je nastavena, jeden bajt udává počet podoblastí, na které je daná oblast rozdělena. Pokud není nastavena, žádné podoblasti neexistují.
- **SubrectsColoured** – pokud je nastavena, každá podoblast je popsána barvou podoblasti, pozicí *x*, *y* a šířkou a výškou dané podoblasti. Pokud není nastavena, všechny podoblasti mají stejnou barvu, barvu popředí.

2.4.5 ZRLE kódování

ZRLE znamená Zlib Run-Length Encoding. Zlib je speciální knihovna vytvořená pro tento typ kódování. Toto kódování je poměrně složité a nad rámec tohoto textu, kombinuje zlib kompresy, „dláždění“, paletizaci a run-length kódování. Bližší podrobnosti jsou k nalezení v [22] a [12].

2.5 Ukončení relace

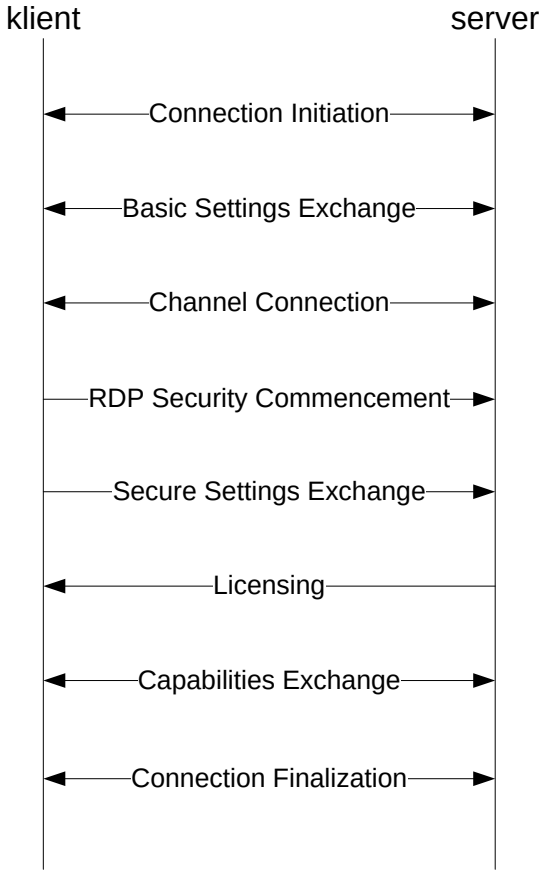
Jak již plyne z návrhu protokolu RFB, ukončení bude velmi jednoduchý proces, jelikož klient si neuchovává vůbec žádné informace. Všechny informace jsou uchovávány serverem. Klientovi stačí k ukončení relace pouhé odpojení od serveru. V případě chyby serveru může iniciovat ukončení relace i server samotný, aniž by to jakkoli narušilo klienta. Plocha je vždy zanechána v takovém stavu, v jakém ji klient zanechal.

3 Protokol RDP

Protokol RDP (Remote Desktop Porotocol) pro řízení vzdálené stanice pochází z dílny firmy Microsoft. Je rozšířením standardu ITU-T T.128. RDP klienti jsou podporováni mnoha platformami (MS Windows, MS Windows Mobile, Linux, Unix, Mac OS). Server ve výchozím nastavení naslouchá na portu TCP 3389. Vzhledem k tomu, že se jedná o dosti složitý protokol, jehož podrobný popis není náplní této práce, pro získání více informací doporučuji [15], [21], [22].

3.1 Sestavení spojení

RDP uskutečňuje spojení na principu klient – server. Sestavení spojení je při použití protokolu RDP velmi složité. Klient se serverem si při něm vymění 23 zpráv rozdělených do osmi kategorií. Těchto osm kategorií, které znázorňuje obr. 3.1, zde bude zevrubně popsáno. Podrobnější popis sestavení spojení je uveden v [15].



Obr. 3.1: Osm kategorií sestavování RDP spojení

Fáze Connection Initiation

Klient zahajuje spojení posláním zprávy X.224 Conection Request PDU (Protocol Data Unit) směrem k serveru. Server odpovídá zprávou X.224 Connection Confirm PDU.

Od tohoto okamžiku jsou všechna data zasílaná mezi klientem a serverem zapouzdřena v X.224 Data PDU, více viz [15].

Fáze Basic Settings Exchange

Základní nastavení mezi klientem a serverem je uskutečněno výměnou zpráv MCS Connect Initial PDU a MCS Connect Response PDU. Connect Initial PDU obsahuje GCC (Generic Conference Control), zatímco Connect Response obsahuje GCC Conference Create Response. Tyto dva GCC pakety obsahují bloky dat s nastavením (např. základní údaje, zabezpečení dat, atd.), která jsou klientem a serverem přečteny, viz [15].

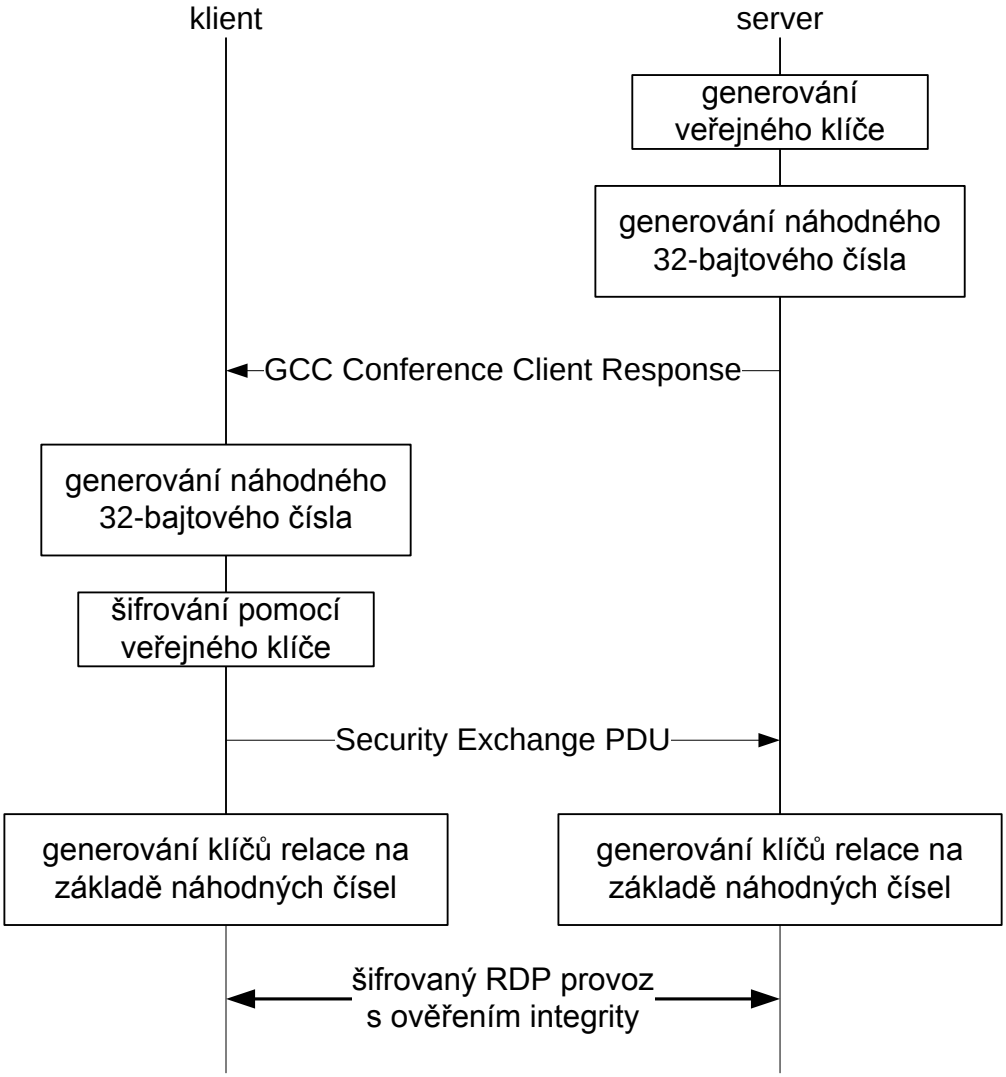
Fáze Channel Connection

Klient odešle zprávy MCS Erect Domain Request PDU a MCS Attach User Request PDU pro připojení klienta k MCS doméně. Server odpovídá zprávou MCS Attach User Confirm PDU, která obsahuje ID uživatelského kanálu. Klient se poté připojuje na různé druhy kanálů pomocí zpráv MCS Channel Join Request PDU. Server potvrzuje každý kanál zprávou MCS Channel Join Confirm PDU.

Od tohoto okamžiku se data zasílaná klientem na server zapouzdřují do MCS Send Data Request PDU. Data posílaná serverem klientovi se zapouzdřují do MCS Send Data Indication PDU. Dále jsou ještě oba druhy dat zapouzdřeny do protokolového komponentu X.224 Data PDU, viz [15].

Fáze RDP Security Commencement

Při použití mechanismu Standard RDP Security (viz 3.3) a při platném šifrování (určeno z údajů v GCC Conference Create Response) odešle klient zprávu Security Exchange PDU obsahující náhodné 32-bajtové číslo serveru. Náhodné číslo je zašifrováno pomocí veřejného klíče. Veřejný klíč, stejně jako 32-bajtové serverem generované náhodné číslo, jsou získány z údajů v GCC Conference Create Response. Klient a server poté využijí tyto dvě 32-bajtová náhodná čísla pro vygenerování klíčů relace použitých k šifrování a ověření integrity následného RDP provozu, viz obrázek 3.2.



Obr. 3.2: Fáze RDP Security Commencement

Od tohoto okamžiku mohou být data šifrována. Bezpečnostní hlavičky jsou součástí dat. Za nimi následují X.224 a MCS záhlaví, která označují, zda jsou data šifrována. V případě platného šifrování jsou data od klienta k serveru šifrována vždy. V opačném směru nemusí být data vždy šifrována, více viz [15].

Fáze Secure Settings Exchange

Tajná data o klientovi (především heslo) jsou odeslána na server pomocí zprávy Client Info PDU, viz [15].

Fáze Licensing

Cílem výměny licencí je převedení licence ze serveru na klienta. Klient uloží licenci a následně ji odešle zpět na server k ověření. V některých případech nemusí být licence klientovi vydána. Výměna paketů v této fázi protokolu závisí na vytíženosti serveru, viz [15].

Fáze Capabilities Exchange

Server odešle výčet svých schopností ve zprávě Demand Active PDU. Klient odpoví výčtem svých schopností ve zprávě Confirm Active PDU. Více viz [15].

Fáze Connection Finalization

V poslední fázi jsou dohodnuty detaily spojení. Zprávy zasílané klientem serveru nemají žádnou závislost na zprávách poslaných serverem klientovi. Mohou být odeslány i v jedné skupině při zachování pořadí, viz [15].

Zprávy od klienta k serveru

V konečné fázi sestavování spojení posílá nejprve klient směrem k serveru pět zpráv:

- Client Synchronize PDU
- Client Control (Cooperate) PDU
- Client Control (Request Control) PDU
- Persistent Key List PDU
- Font List PDU

Zprávy od serveru ke klientovi

Po obdržení zpráv od klienta zasílá server poslední čtyři zprávy, kterými je sestavení spojení dokončeno:

- Server Synchronize PDU
- Server Control (Cooperate) PDU
- Server Control (Granted Control) PDU
- Font Map PDU

3.2 Podporované vlastnosti

Protokol RDP podporuje celou řadu funkcí, z nichž velká většina je použita v existujících řešeních, která jsou popsána v podkapitole Existující řešení řízení vzdálené stanice.

Podporuje:

- barevnou hloubku 8, 15, 16, 24 a 32 bitů,
- Audio Redirection – přesměrování výstupu zvuků umožňuje poslech zvuků ze serveru na lokálním počítači,
- File System Redirection – použití lokálního souboru na vzdálené stanici,
- Printer Redirection – použití lokální tiskárny pro tisk souborů umístěných na vzdálené stanici
- Port Redirection – použití lokálních sériových a paralelních portů pro programy spuštěné na vzdálené stanici,
- schránku pro kopírování textu mezi lokální a vzdálenou stanicí,
- Remote programs – asociování programů ze vzdálené stanice se soubory umístěnými na lokální stanici,
- Seamless Windows – vzdálené aplikace mohou běžet na stanici klienta, jako by byly spuštěny lokálně,
- Terminal Server Gateway – připojení prostřednictvím HTTPS portu 443,
- grafické rozhraní Windows Aero,
- Multiple session – zobrazování plochy na více monitorech, viz [21].

3.3 Bezpečnostní funkce

Vyjednávání o zabezpečení probíhá následovně. Klient zašle jím podporované šifrovací úrovně ve zprávě Client Security Data. Na základě obdržených dat server určí úroveň šifrování a odešle ji zpět klientovi ve zprávě Server Security Data. Klient na základě obdržené zprávy nakonfiguruje své kryptografické moduly. Pokud server není schopen vybrat žádnou z úrovní šifrování, jednání selže a musí být ukončeno připojení k síti, viz [15].

Základem bezpečnostních funkcí protokolu RDP je Standard RDP Security. Ten podporuje čtyři úrovně šifrování:

- Low – všechna data odesílaná od klienta směrem k serveru jsou chráněna šifrováním založeným na maximální síle klíče podporované klientem,
- Client Compatible – všechna data posílaná mezi klientem a serverem jsou chráněna šifrováním založeným na maximální síle klíče podporované klientem,
- High – všechna data posílaná mezi klientem a serverem jsou chráněna šifrováním založeným na maximální síle klíče podporované serverem,
- FIPS – všechna data posílaná mezi klientem a serverem jsou chráněna pomocí Federal Information Processing Standard 140-1.[3]

[3] FIPS 140-1 - standard pro počítačovou bezpečnost americké vlády. Specifikuje požadavky pro kryptografické moduly.

4 Implementace bezpečného řízení vzdálené stanice

V této stěžejní kapitole celé práce je podrobně rozebrána výsledná implementace bezpečného řízení vzdálené stanice. Nejprve je obecně popsán význam a použití aplikace včetně jednotlivých možností nastavení. Poté jsou podrobně popsány zabezpečovací funkce aplikace, jejichž implementace byla úkolem této práce. V další části jsou provedeny analýzy jednotlivých typů šifrovaní. V poslední části kapitoly jsou uvedeny náměty na další možnost rozšíření práce.

4.1 Obecný popis aplikace

Základní účel aplikace spočívá v bezpečném řízení vzdálené stanice. Míru bezpečnosti určuje sám uživatel v podobě volby vhodného typu autentizace a šifrování přenášených dat. Šifrován je i samotný přenos vzdálené plochy. Při změně typu šifrování je nutné celou aplikaci restartovat. Tato nutnost vyplývá z použití externího provideru. Aplikace dále nabízí přenos souborů mezi počítači a možnost kopírování textu pomocí schránky.

Jako základ aplikace byl využit open source projekt Java Remote Desktop verze 0.3.1.0 licencovaný pod GPL (GNU General Public License). Z názvu je patrné, že se jedná o implementaci v programovacím jazyce Java. Komunikaci mezi klientskou a serverovou aplikací zde zprostředkovává Java RMI API (Java Remote Method Invocation Application Program Interface). Jedná se o rozhraní jazyka Java umožňující volat metody objektů umístěných na serveru. Volání metod na vzdálených objektech je téměř shodné s voláním metod na objektech lokálních. Podrobnější popis Java RMI API lze nalézt v [18].

Vzhledem k tomu, že byl použit programovací jazyk Java, je možné aplikaci spouštět na všech dostupných platformách (MS Windows, Linux, Mac OS, Solaris OS). Aplikaci je možné spouštět jak v podobě klasické aplikace, tak v podobě webového appletu. Spouštěcí soubor appletu se nachází ve složce SpousteciSoubory na přiloženém CD. Applet je též možné spustit v prostředí NetBeans IDE pomocí třídy `mainApplet.java` umístěné v balíčku `jrdesktop`.

Ke spuštění aplikace slouží dva JAR archivy umístěné ve složce SpousteciSoubory na přiloženém CD. Spuštění aplikace je nutné provést z příkazové řádky zadáním příkazu:

java -cp bcprov-jdk16-146.jar;jrdesktop.jar jrdesktop.main – pro systém MS Windows,
java -cp bcprov-jdk16-146.jar:jrdesktop.jar jrdesktop.main – pro systém Linux.

Rozdíl ve spouštěcích příkazech je pouze ve středníku před názvem souboru jrdesktop.jar, který je použit v příkazu pro spouštění aplikace v systému MS Windows. Pro spuštění aplikace v systému Linux je na místě středníku použita dvojtečka.

Pro test aplikace je nutné použít JRE 1.6 (Java Runtime Environment), viz [17]. Pro správnou funkci všech délek klíčů šifrovacího algoritmu AES je nutné nahradit výchozí soubory soubory z archivu JCE Unlimited Strength Jurisdiction Policy File 6 a to jak na straně serveru, tak na straně klienta. Přesný postup je blíže popsán v podkapitole 4.2.5 v části AES encryption.

Vzhledem k tomuto problému je nutné po každé změně šifrování restartovat celou aplikaci. Veškerá jiná nastavení je možné měnit za běhu aplikace.

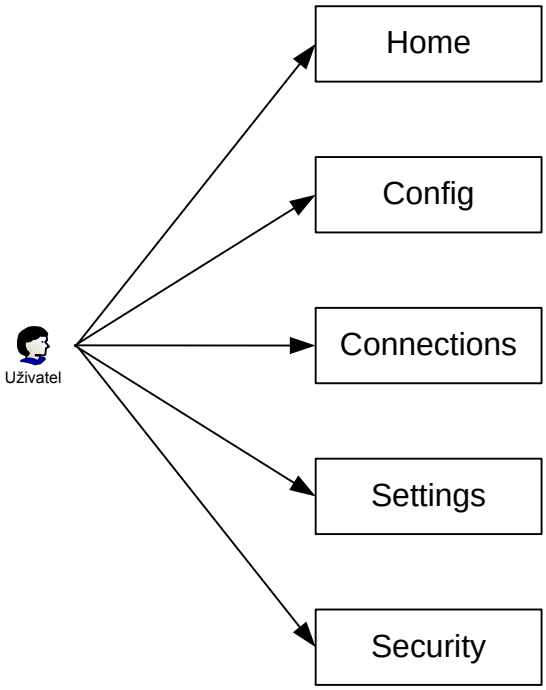
Aplikace byla testována na systémech MS Windows XP Professional SP3, MS Windows 7 Professional, Ubuntu 10.10 a Ubuntu 11.04. Na všech zmíněných systémech byla aplikace plně funkční.

Za největší nevýhodu výsledné aplikace lze považovat její funkčnost pouze v místní síti (LAN), jelikož nezvládá práci s překladem adres (NAT). Při případném rozšiřování aplikace by bylo vhodné tento nedostatek odstranit jako první.

4.2 Možnosti nastavení

Základní okno aplikace je stejné pro klientskou i serverovou část. Pro spuštění serverové aplikace slouží tlačítko **Start**. Tlačítko **New connection…** spouští aplikaci klientskou. Jednotlivé možnosti nastavení jsou popsány v této podkapitole. Nastavení určuje sám uživatel.

Po spuštění aplikace má uživatel na výběr z pěti záložek obsahujících různá nastavení, viz obrázek 4.1. Názvy záložek byly voleny tak, aby orientace v aplikaci byla co nejvíce intuitivní.



Obr. 4.1: Základní rozdělení nastavení

4.2.1 Záložka Home

Tato záložka nabízí tři tlačítka (**Start**, **New connection…** a **Exit**) a okno **Status**. V okně **Status** jsou po připojení k síti vypsány základní parametry, jako IP adresa, port, zvolený druh autentizace a šifrování, apod.

Tlačítkem **Start** se spustí serverová část aplikace s parametry nastavenými v záložkách **Config** (IP adresa, port) a **Security** (typ autentizace a šifrování).

Tlačítkem **New connection…** se otevře nové okno. To slouží pro připojení klientské části aplikace na serverovou. Obsahuje nastavení konfigurace připojení. Konfiguraci lze nastavit některou již dříve použitou výběrem příslušného názvu v okně **Configuration**, případně použít jinou konfiguraci, kterou je nutné nastavit manuálně. Je zde nutné nastavit IP adresu a port serveru, na který se chceme připojit. Dále je v poli **Authentication** možné měnit uživatelské jméno a heslo. Tyto parametry lze měnit jen při určitých typech autentizace, kdy se využívají. Uživatelské jméno i heslo musí odpovídat nastavení na serveru, jinak se nebude možné autentizovat a spojení se s chybovým hlášením ukončí. Dále je zde ještě možné zvolit **Connection mode** jako **Secured connection** (šifrování pomocí SSL a použití serverového i klientského certifikátu pro vzájemnou autentizaci) a **Reverse connection** (prohození rolí klienta a serveru). Tlačítko **Exit** slouží ke standardnímu ukončení aplikace.

4.2.2 Záložka Config

Tato záložka obsahuje nastavení konfigurace pro serverovou část aplikace. Konfiguraci lze opět vybrat z dříve použitých výběrem příslušného názvu, případně vytvořit novou manuálně.

Pole **IP Address** slouží k nastavení IP adresy serveru, což je vlastně IP adresa síťové karty, na kterou se bude klient připojovat. Menu **Address** poskytuje výběr ze všech aktuálních IP adres včetně adresy tzv. localhosta. Adresa localhosta slouží ovšem pouze pro testování, jelikož připojovat se vzdáleně na počítač, se kterým právě pracujeme, není praktické. Nad menu **Address** jsou tři položky checkboxu, které slouží k manuálnímu výběru adresy, automatickému použití zvolené adresy a použití zvolené adresy ze seznamu jako výchozí.

Pole **Ports** slouží k nastavení portů pro klasickou aplikaci a pro webový applet. Výchozí nastavení pro port klasické aplikace je 1099 a pro applet 6666.

Pole **Connection mode** a **Authentication** mají stejný význam jako v okně **New connection…** s tím rozdílem, že zde jsou nastavené parametry použity pro serverovou část aplikace.

4.2.3 Záložka Connections

Tato záložka má význam pouze u serverové aplikace. Zobrazují se zde všechna aktivní připojení na server. Má spíše informativní charakter, jediné, co zde lze provést je ukončení všech nebo vybraných připojení.

Po stisku tlačítka **Properties** se zobrazí okno s vlastnostmi připojení, jako jsou název klientského počítače a jeho IP adresa. Dále používanou verzi Javy a používaný operační systém, uživatelské jméno, adresář, odkud je aplikace spuštěna, a rozlišení obrazovky.

Po stisku tlačítka **Details** se lze dozvědět vlastnosti připojení. Vlastnosti jsou následující: doba připojení, velikost odeslaných a doručených dat, celkový objem dat, přenosová rychlost, typ autentizace a šifrování. Do tohoto výpisu byly přidány měřící funkce: počet požadavků směrem od klienta k serveru, počet požadavků směrem serveru ke klientovi, průměrná doba zpracování jednoho požadavku v obou směrech a průměrná velikost jednoho požadavku v obou směrech.

4.2.4 Záložka Settings

Tato záložka umožňuje nastavit vzhled okna aplikace, na výběr je z pěti možností a dále například skrývání hlavního okna aplikace. Dále je zde možné nastavit proxy server. Ve spodní části okna je možné zvolit adresář, do kterého se budou ukládat stahované soubory. Nachází se zde i tlačítko **Clear configuration files**, které slouží k uvedení aplikace do výchozího nastavení.

4.2.5 Záložka Security

Na této záložce má uživatel možnost nastavit různé stupně ochrany. Je zde na výběr ze čtyř typů autentizace a šifrování. Vzhledem k tomu, že implementace bezpečnostních funkcí byla cílem této diplomové práce, bude této záložce věnována největší pozornost. Pro implementaci bezpečnostních funkcí byly převážně použity standardní kryptografické knihovny, které jsou součástí JDK (Java Development Kit).

None authentication

Tato volba znamená režim bez autentizace uživatele. Režim bez autentizace není vhodné používat, jelikož se na server může připojit prakticky kdokoli. Případnému útočníkovi stačí pouze zjistit na jaké IP adrese a portu je server spuštěn a má neomezený přístup do serverového počítače. Pokud se na něm nachází nějaká citlivá data, tak to může mít nedozírné následky, protože útočníkovi nic nebrání v tom data použít, případně odstranit.

Basic authentication

Autentizace pomocí uživatelského jména a hesla neboli tzv. autentizace znalostí. Jedná se o v dnešní době nejběžnější typ autentizace. Je založena na shodě zhašovaných sdílených tajemstní serverové a klientské části aplikace. Server vypočte haš z uživatelského jména a hesla, které je na něm nastaveno. Klientská aplikace též vypočte haš ze svého uživatelského jména a hesla. Vypočtený haš poté odešle na server. Server porovná svůj vypočtený haš s přijatým. Pokud se oba haše shodují, autentizace proběhla v pořádku. Pokud ne, klientovi není umožněn přístup na server a je mu odeslána chybová hláška o selhání autentizace.

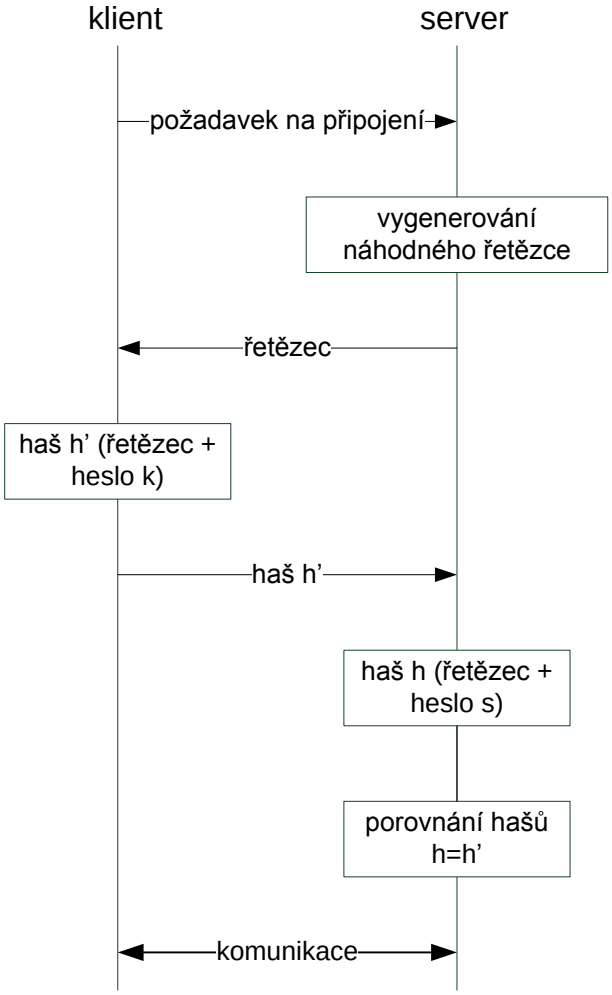
OTP authentication (One-Time Password)

Autentizace pomocí jednorázového hesla. Klient zasílá serveru časovou značku (timestamp, token) zašifrovanou heslem klienta. Server ji svým heslem rozšifruje, hesla se samozřejmě musí opět shodovat, a zapíše si ji do svého blacklistu. Tím je zajištěno, že se povolí přístup pouze jednomu klientovi s tímto tokenem. Server umožňuje nastavit dobu platnosti tohoto

hesla. Po vypršení platnosti není klientovi umožněn přístup na server a je mu odeslána chybová hláška o vypršení platnosti tokenu. V základním nastavení je doba platnosti tokenu šedesát sekund.

Phased authetication

Autentizace typu výzva-odezva. Klient posílá serveru požadavek na připojení. Server pošle klientovi náhodně vygenerovaný řetězec znaků a uloží si ho. Klient k řetězci přidá své heslo a řetězec i s heslem zhašuje pomocí algoritmu SHA. Haš poté odešle zpět serveru. Server vezme dle IP adresy klienta jemu příslušející dříve uložený náhodný řetězec. Spojí ho se svým heslem a zhašuje za pomoci algoritmu SHA. Oba haše jsou poté spolu porovnány, a pokud se shodují, znamená to, že klient zná správné heslo a autentizace proběhla v pořádku. Klientovi je poté umožněn přístup na server.



Obr. 4.2: Phased authentication

DES encryption

Pro DES šifrování je stěžejní balíček JCE (Java Cryptographic Extension). Jeho jádrem je třída `javax.crypto.Cipher`. Samotné šifrování a dešifrování je starostí přímo této třídy. Pro práci s 56 bitů dlouhým šifrovacím klíčem je poté využito rozhraní `javax.crypto.spec.DESKeySpec`. Šifra DES je v této aplikaci použita v módu CFB (Cipher FeedBack). Tento mód znamená využití zpětné vazby z předchozího šifrovaného bloku. Je realizována pomocí posuvného registru.

AES encryption

Pro šifrování AES není možné použít `SecretKeyFactory` jako tomu bylo u šifry DES. Místo toho byla využita třída `javax.crypto.spec.SecretKeySpec`. Mód byl použit opět CFB.

Pro správnou funkci šifrování AES bylo nutné použít externí security provider. Byl zvolen Bouncy Castle (BC) provider, viz [3]. Ten je k aplikaci dodán ve formě externí JAR (Java Archive) knihovny.

Problém u šifrování AES nastal s délkami klíčů 192b a 256b. Tyto délky klíčů nejsou ve standardní Javě podporovány. Pro správnou funkci klíčů je nutné stáhnout ze stránek Oracle [17] JAR soubory JCE Unlimited Strength Jurisdiction Policy File 6. Těmito soubory je nutné nahradit stávající soubory v adresáři <JAVA_ROOT>/jre/lib/security. Přepsání souborů je nutné provést jak na straně serveru, tak na straně klienta. Po přepsání stávajících souborů na obou stranách komunikace je již možné délky klíčů 192b a 256b u šifrování AES využít.

RC4 encryption

Pro implementaci této proudové šifry byly použity standardní třídy, jako v případě šifry DES. Pro šifrování a dešifrování je využita třída `javax.crypto.Cipher`. Pro práci s klíčem je poté opět využita třída `javax.crypto.spec.SecretKeySpec`. Pro tuto aplikaci byla zvolena délka klíče proudové šifry RC4 128 bitů.

4.2.6 Okno klienta

Okno klientské části aplikace nabízí též mnohá nastavení. Nachází se zde tlačítko **Stop** a **Pauza**, které pozastavují přenos obrazu plochy serveru. Přenos obrazu lze opět obnovit kliknutím na tlačítko **Play**. Dále se zde nachází tlačítko, které přepíná mezi režimem ovládání

a pouhého sledování vzdálené plochy serveru. Následující tlačítko umožňuje přepínat mezi normálním režimem a režimem celé obrazovky. Další čtyři možnosti upravují velikost a kvalitu přenášeného obrazu. Následující položka umožňuje nastavit, po kolika milisekundách se má obraz vzdálené plochy obnovovat. Na výběr je zde z osmi možností v rozmezí 0 až 9000 milisekund. Následují tlačítka pro povolení kopírování textu pomocí schránky, umožňující přenos souborů a tlačítko nápovědy, které obsahuje i informace o připojení stejné jako v záložce **Connections**. Poslední tlačítko slouží ke standardnímu ukončení klientského okna a návrat do základního okna aplikace.

4.3 Analýza použitých typů šifrování

V této podkapitole bude popsána analýza implementovaných šifrování. Analýza byla prováděna za použití stejného scénáře pro všechny druhy šifrování. Bylo zde navíc rozlišeno, zda byla použita obrazová komprese, či nikoli.

Scénář byl následující: Nejprve bylo navázáno spojení se serverem. Po navázání spojení bylo okno klientské aplikace maximalizováno na celou obrazovku. Na serveru byl z plochy spuštěn poznámkový blok. Do poznámkového bloku byla napsána věta: „Toto je analýza šifrování diplomové práce.“ Po zobrazení celé věty byl poznámkový blok minimalizován. Poté byl z nabídky Start spuštěn internetový prohlížeč a ten byl následně maximalizován na celou plochu. Po kompletním načtení domovské stránky, byl prohlížeč minimalizován a scénář ukončen. Domovská stránka prohlížeče byla nastavena na statickou internetovou stránku, aby byly výsledky měření co nejprůkaznější. V okně klienta byly zobrazeny výsledky měření. Analýza byla prováděna při rozlišení obrazovky 1280x720 obrazových bodů a při obnově obrazu vzdálené plochy každých 500ms. Výsledky měření efektivity implementace zabezpečeného řízení vzdálené stanice při různých typech šifrování jsou zobrazeny v tabulce 4.1 s použitím obrazové komprese a v tabulce 4.2 bez použití obrazové komprese.

Tab. 4.1: Výsledky měření efektivity implementace zabezpečeného řízení vzdálené stanice při různých typech šifrování s použitím obrazové komprese

		s použitím obrazové komprese					
algoritmus		None	DES	AES	AES	AES	RC4
délka klíče		-	56b	128b	192b	256b	128b
událost							
celkový objem dat S -> K	[MB]	1,86	1,90	2,14	1,85	1,87	1,80
celkový objem dat K -> S	[kB]	94,33	101,46	84,52	83,12	72,96	91,92
počet požadavků S -> K	[-]	84,00	90,00	88,00	81,00	81,00	85,00
počet požadavků K -> S	[-]	56,00	60,00	60,00	54,00	54,00	58,00
průměrná doba zpracování požadavku S -> K	[ms]	248,00	232,00	240,00	249,00	255,00	243,00
průměrná doba zpracování požadavku K -> S	[ms]	61,00	50,00	45,00	51,00	50,00	46,00
průměrná velikost požadavku S -> K	[kB]	22,76	21,64	24,91	23,43	23,71	21,70
průměrná velikost požadavku K -> S	[kB]	1,68	1,69	1,40	1,53	1,35	1,58

Tab. 4.2: Výsledky měření efektivity implementace zabezpečeného řízení vzdálené stanice při různých typech šifrování bez použití obrazové komprese

		bez použití obrazové komprese					
algoritmus		None	DES	AES	AES	AES	RC4
délka klíče		-	56b	128b	192b	256b	128b
událost							
celkový objem dat S -> K	[MB]	3,21	3,22	3,15	3,35	3,07	2,93
celkový objem dat K -> S	[kB]	85,81	81,40	83,20	92,63	81,09	83,09
počet požadavků S -> K	[-]	96,00	97,00	96,00	102,00	96,00	91,00
počet požadavků K -> S	[-]	64,00	66,00	65,00	68,00	64,00	62,00
průměrná doba zpracování požadavku S -> K	[ms]	159,00	150,00	146,00	149,00	147,00	148,00
průměrná doba zpracování požadavku K -> S	[ms]	40,00	41,00	51,00	40,00	43,00	42,00
průměrná velikost požadavku S -> K	[kB]	34,32	34,02	33,62	33,70	32,85	33,01
průměrná velikost požadavku K -> S	[kB]	1,34	1,23	1,27	1,36	1,26	1,33

Z naměřených hodnot byly poté vypočteny ještě další dvě tabulky, ve kterých je proveden výpočet doby zpracování požadavku o velikosti 20kB a dále výpočet objemu zpracovaných dat za 1 sekundu. V těchto výpočtech byl uvažován směr od serveru ke klientovi. Vypočtené výsledky zobrazují tabulky 4.3 a 4.4.

Tab. 4.3: Vypočtené výsledky s použitím obrazové komprese

		s použitím obrazové komprese					
algoritmus		None	DES	AES	AES	AES	RC4
délka klíče		-	56b	128b	192b	256b	128b
událost							
průměrná doba zpracování požadavku o velikosti 20kB	[ms]	217,93	214,42	192,69	212,55	215,10	223,96
průměrná velikost dat zpracovaných za 1s	[kB]	91,77	93,28	103,79	94,10	92,98	89,30

Tab. 4.4: Vypočtené výsledky bez použití obrazové komprese

		bez použití obrazové komprese					
algoritmus		None	DES	AES	AES	AES	RC4
délka klíče		-	56b	128b	192b	256b	128b
událost							
průměrná doba zpracování požadavku o velikosti 20kB	[ms]	92,66	88,18	86,85	88,43	89,50	89,67
průměrná velikost dat zpracovaných za 1s	[kB]	215,85	226,80	230,27	226,17	223,47	223,04

Z naměřených a vypočtených hodnot je patrné, že při použití obrazové komprese klesne objem dat přenesených od serveru ke klientovi zhruba o třetinu. Průměrná doba zpracování jednoho požadavku je ovšem bez použití komprese obrazu o cca 35% kratší. Dále z vypočtených hodnot vyplývá, že použití šifrovacího algoritmu nemá na dobu zpracování požadavku téměř žádný vliv, rozdíly jsou pouze minimální. Přesto z tabulek 4.3 a 4.4 vyplývá, že nejvhodnějším šifrováním implementovaným ve výsledné aplikaci je šifrování AES s délkou klíče 128 bitů. Za jednu vteřinu je schopno zpracovat největší objem dat.

Další provedené analýzy byly založeny změně frekvence obnovy obrazu vzdálené plochy. Tyto analýzy jsou však velmi subjektivní a do jisté míry záleží i na schopnostech uživatele samotného.

První z těchto analýz spočívala v psaní textu v poznámkovém bloku, který byl otevřen na straně serveru. Tuto analýzu prováděla osoba, pro kterou je psaní na počítači pracovní náplní. Za kritickou hodnotu bylo považováno zpoždění dvou celých slov v obraze. Podle hodnotící osoby nemělo vliv, zda byla nebo nebyla použita komprese obrazu. Výsledky byly v obou případech stejné. Výsledky shrnuje tabulka 4.5. Znovu ovšem upozorňuji, že výsledky jsou značně subjektivní.

Tab. 4.5: Analýza psaní textu

algoritmus		None	DES	AES	AES	AES	RC4
délka klíče		-	56b	128b	192b	256b	128b
maximální použitelná doba obnovy	[ms]	500	500	500	500	250	250

Poslední provedenou analýzou je analýza běžné práce na ploše serveru, jako například procházení adresářů, spouštění prohlížeče, apod. Tato analýza poskytuje opět velmi subjektivní výsledky a to, co pro někoho může být ještě přijatelná odezva, pro jiného může být odezva značně nedostačující. Tuto analýzu provádělo pět nezávislých osob pracujících v různých oborech. Nezávisle na sobě se shodli na tom, že při jakémkoli použitém šifrovacím algoritmu je maximální přijatelná doba obnovy vzdálené plochy 1 sekunda. Při této analýze opět nezáleželo na tom, zda byla či nebyla použita obrazová komprese.

4.4 Návrhy na rozšíření

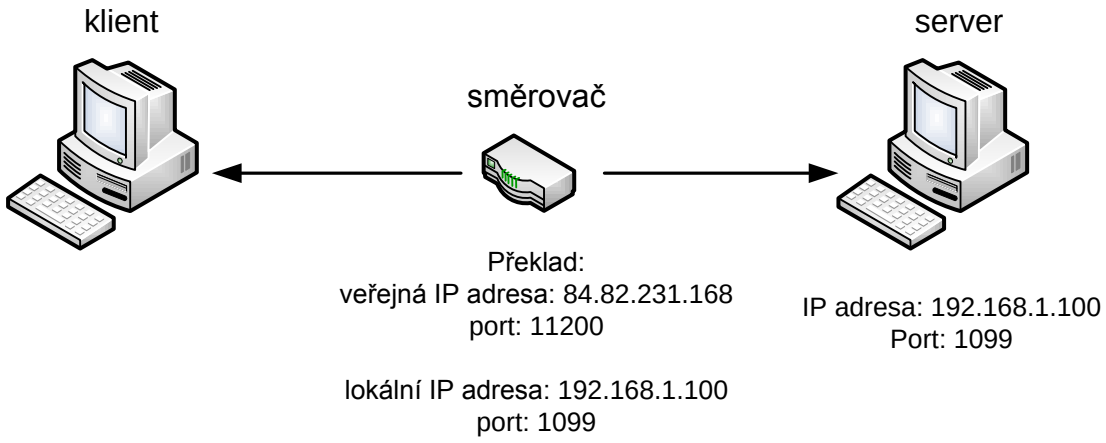
V této podkapitole jsou představeny návrhy na možné rozšíření výsledné aplikace. Rozšíření se nabízí mnoho, zde budou uvedeny jen ty nejpraktičtější.

4.4.1 Práce za NATem

V síti Internet je velmi častým jevem využití techniky nazývané NAT. Jedná se o překlad IP adres, který mění IP adresu zařízení v lokální síti na adresu veřejnou. Prakticky se jedná o to, že za jedním síťovým prvkem, nejčastěji směrovačem, je schováno více počítačů, které na venek vystupují pod jedinou IP adresou. Rozlišení, kterému počítači která komunikace patří se pak děje na základě portů. Pro komunikaci v celé síti Internetu je v dnešní době schopnost práce s NATem nezbytná. Proto by bylo vhodné doimplementovat tuto schopnost jako první. Aplikace s NATem v současnosti pracovat neumí, proto je její funkčnost omezena pouze na lokální síť, kde k žádnému překladu adres nedochází.

Možností řešení práce s NATem existuje několik. Nejjednodušším, a v dnešní praxi i nejběžnějším, je úprava nastavení směrovače tak, aby veškerou komunikaci směřující na zvolený port posílal na vybraný počítač v lokální síti. Tato technika je označována jako přesměrování portů (port forwarding).

Příklad této techniky je uveden na obrázku 4.3.



Obr. 4.3: Příklad techniky NAT

4.4.2 Přenos hlasu

Mnoho dnes již existujících řešení funkci přenosu hlasu využívá. Přenos hlasu může sloužit například k výuce pomocí vzdálené plochy, případně ke komunikaci se zákazníkem při technické podpoře. Je uživatelsky přívětivější než komunikace prostřednictvím textového chatu a z tohoto důvodu je i stále více v podobných aplikacích zaváděn.

Samotnou realizaci by bylo možné řešit pomocí protokolů SIP a RTP. Protokol SIP by měl za úkol vyjednat parametry relace, protokol RTP by byl určen k samotnému přenosu hlasu. Současně s přenosem hlasu by byl přenášen obraz vzdálené plochy. Při implementaci přenosu hlasu by bylo nezbytné provést další změny pro projití NATem, pokud by byl již dříve implementován.

Závěr

Bezpečnostní prvky jsou důležitou součástí řízení vzdálené stanice. Lze je rozdělit do dvou kategorií. První kategorií je autentizace, která zabraňuje neoprávněnému přístupu na stanici. Druhou kategorií je šifrování veškeré komunikace se stanicí. Šifrování zabraňuje nežádoucímu odposlech a případnému zneužití dat získaných útočníkem při jejich přenosu mezi lokální a vzdálenou stanicí. Za předpokladu, že je k šifrování použit kvalitní šifrovací algoritmus je získání původních dat téměř nemožné. Nejbezpečnější způsob ochrany představuje kombinace obou těchto kategorií.

Úvodem se práce zabývá popisem různých využití řízení vzdálených stanic. Jsou zde zhodnoceny jejich výhody a případné nedostatky. Jednou z hlavních výhod je úspora času, které uživatel využitím vzdálených stanic dosáhne. Dále jsou podrobně popsány základní kryptografické prvky, které jsou nezbytné k pochopení celé práce. Některé ze zde uváděných algoritmů jsou použity ve výsledné implementaci bezpečného řízení vzdálené stanice. Závěrečná část úvodní kapitoly se zabývá popisem existujících řešení řízení vzdálených stanic. Existujících řešení existuje celá řada a mohou mezi nimi být markantní rozdíly. Popis každého řešení je pro snazší zhodnocení výsledků rozdělen do tří samostatných kategorií. Ze zde uvedených řešení se jako nejlepší jeví TeamViewer. Nabízí bezesporu nejširší škálu funkcí a pro nekomerční využití je v plné verzi zcela zdarma.

Následuje detailní popis protokolu RFB. Jedná se o velmi jednoduchý a dobře specifikovaný protokol. Pro implementaci při řízení vzdálené stanice je však zcela postačující.

Práce se též zabývá popisem protokolu RDP. Jedná se o velmi složitý protokol z dílny firmy Microsoft. Proto je i jeho popis dosti zestručněn.

V dnešní době je v praxi více vyžívaný protokol RFB. Ten je též použit v převážné většině open source projektů. V placených aplikacích je často využíván proprietální protokol, který si firmy tvoří samy na míru.

Poslední kapitola je pro tuto práci stěžejní a obsahuje popis implementace bezpečného řízení stanice. Nejprve je uveden stručný obecný popis aplikace. Pro implementaci byl zvolen programovací jazyk Java, čímž je zaručena přenositelnost aplikace mezi všemi dnes používanými platformami. Jako základ aplikace byl využit open source projekt Java Remote

Desktop verze 0.3.1.0 licencovaný pod GPL. Do něj byly naimplementovány bezpečnostní prvky (vybrané typy autentizace a šifrování).

Implementace obsahuje několik záložek s různými možnostmi nastavení jak serverové, tak klientské aplikace. Základní okno je pro serverovou i klientskou aplikaci stejné, zda jde o první či druhý typ aplikace se rozliší finálním spuštěním po nastavení požadovaných parametrů. Největší pozornost je věnována záložce **Security**, jejíž implementace byla úkolem této diplomové práce. Tato záložka nabízí možnost volby ze čtyř druhů autentizace a šesti druhů šifrování. Jednotlivé možnosti lze vzájemně libovolně kombinovat. U jednotlivých typů autentizace a šifrování je popsán jejich princip a popis jejich implementace.

Dále byla provedena analýza jednotlivých druhů šifrování. Výsledky jednotlivých analýz jsou pro lepší přehlednost umístěny v tabulkách. Některé z provedených analýz jsou značně subjektivně závislé. Veškeré analýzy ovšem ukázaly, že na zvoleném šifrovacím algoritmu téměř nezáleží. Rozdíly mezi nimi jsou minimální. Jako nejefektivnější se přesto ukázal algoritmus AES s délkou klíče 128 bitů. Proto bych z hlediska bezpečnosti a efektivity doporučoval nastavovat kombinaci Phased autentizace a šifrování AES s délkou klíče 128 bitů.

Nakonec byly nabídnuty podněty na možná rozšíření aplikace. Jelikož výsledná aplikace je určena pro práci v LAN, jako první je zde uveden popis pro rozšíření působnosti na celou síť Internet. Bylo by nutné implementovat mechanizmy pro schopnost práce s překladem adres, tzv. NATem. Nejběžněji používaným způsobem je úprava nastavení směrovače. Tato technika se nazývá port forwarding. Další zde předkládaná možnost rozšíření je přenos hlasu. Mnoho dnes existujících aplikací přenos hlasu využívá, jelikož vzájemná komunikace dvou stran je mnohem rychlejší než při využití např. textového chatu. Samotný přenos hlasu by byl realizován pomocí protokolů SIP a RTP.

Seznam použitých zdrojů

[1] Advanced Encryption Standard. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 20. 3. 2007, last modified on 20. 10. 2010 [cit. 2010-11-16]. Dostupné z WWW: <http://cs.wikipedia.org/wiki/Advanced_Encryption_Standard>.

[2] Autentizace. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 9. 4. 2008, last modified on 12. 8. 2010 [cit. 2010-12-11]. Dostupné z WWW: <http://cs.wikipedia.org/wiki/Autentizace>.

[3] *Bouncy Castle* [online]. 2001, 24.2.2011 [cit. 2011-05-14]. The Legion of the Bouncy Castle. Dostupné z WWW: <http://www.bouncycastle.org/java.html>.

[4] BURDA, Karel. *Bezpečnost informačních systémů*. Brno : [s.n.], 2005. 104 s.

[5] Comparison of remote desktop software. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 3 November 2010 , last modified on 10 December 2010 [cit. 2010-12-11]. Dostupné z WWW: <http://en.wikipedia.org/wiki/Comparison_of_remote_desktop_software>.

[6] Framebuffer. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 2. 4. 2008, last modified on 6. 1. 2010 [cit. 2010-11-08]. Dostupné z WWW: <http://cs.wikipedia.org/wiki/Framebuffer>.

[7] *GoToMyPC* [online]. 1997, 2010 [cit. 2010-11-15]. Remote Access. Dostupné z WWW: <http://www.gotomypc.eu/remote_access/remote_access>.

[8] HOYTE, Doug. *Hardcore Sofrware* [online]. 2000 [cit. 2010-12-11]. Challenge-Response Authentication. Dostupné z WWW: <http://www.hcsw.org/reading/chalresp.txt>.

[9] KLÍMA, Vlastimil. *Crypto-world* [online]. 5. 4. 2005 [cit. 2010-11-16]. Symetrická kryptografie II. Dostupné z WWW: <http://crypto-world.info/klima/mffuk/Symetricka_kryptografie_II_2006.pdf>.

[10] Kryptologie: Šifrovací algoritmus DES. In *Specialista* [online]. [s.l.] : [s.n.], 1. 1. 2006 [cit. 2010-11-16]. Dostupné z WWW: <http://magazin.specialista.info/view.php?cisloclanku=2006010201>.

[11] *LogMeIn* [online]. 2003, 2010 [cit. 2010-11-15]. Remote Access and Desktop Control Software for Your Computer. Dostupné z WWW: <https://secure.logmein.com/>.

[12] MALINA, L. *Ochrana soukromí na Internetu.* Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2010. 75 s. Vedoucí diplomové práce Ing. Jan Hajný [cit. 2010-12-11]. Dostupné z WWW: <https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=26637&lang=0>

[13] Man in the middle. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 8. 1. 2009, last modified on 7. 7. 2010 [cit. 2010-11-17]. Dostupné z WWW: <http://cs.wikipedia.org/wiki/Man_in_the_middle>.

[14] MD5. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 21 June 2010 , last modified on 12 November 2010 [cit. 2010-11-17]. Dostupné z WWW: <http://en.wikipedia.org/wiki/MD5>.

[15] *MSDN | Microsoft Development, Subscriptions, Resources, and More* [online]. 22. 2. 2007, 19. 11. 2010 [cit. 2010-12-01]. Remote Desktop Protocol: Basic Connectivity and Graphics Remoting Specification. Dostupné z WWW: <http://msdn.microsoft.com/en-us/library/cc240445%28v=PROT.10%29.aspx>.

[16] *MSRC4 DSM Plugin for UltraVNC - Overview* [online]. 2004, 2. 11. 2005 [cit. 2010-11-17]. MSRC4 Plugin Overview. Dostupné z WWW: <http://msrc4plugin.home.comcast.net/~msrc4plugin/overview.html>.

[17] *Oracle.com* [online]. 1994, 4.5.2011 [cit. 2011-05-14]. Java SE downloads. Dostupné z WWW: <http://www.oracle.com/technetwork/java/javase/downloads/index.html>.

[18] *Oracle.com* [online]. 1994, 16.7.2010 [cit. 2011-05-14]. Remote Method Invocation Home. Dostupné z WWW: <http://www.oracle.com/technetwork/java/javase/tech/index-jsp-136424.html>.

[19] RC4. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 4. 10. 2010, last modified on 6. 11. 2010 [cit. 2010-11-17]. Dostupné z WWW: <http://cs.wikipedia.org/wiki/RC4>.

[20] RC6. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 27 October 2005 , last modified on 14 October 2010 [cit. 2010-11-29]. Dostupné z WWW: <http://en.wikipedia.org/wiki/RC6>.

[21] Remote Desktop Protocol. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 30. 12. 2008, last modified on 21. 9. 2010 [cit. 2010-11-30]. Dostupné z WWW: <http://cs.wikipedia.org/wiki/Remote_Desktop_Protocol>.

[22] Remote Desktop Protocol. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 13 December 2009, last modified on 24 November 2010 [cit. 2010-11-30]. Dostupné z WWW: <http://en.wikipedia.org/wiki/Remote_Desktop_Protocol>.

[23] RICHARDSON, Tristan. *The RFB Protocol* [online]. [s.l.] : [s.n.], 2007, Last updated 26 November 2010 [cit. 2010-10-07]. Dostupné z WWW: <http://www.realvnc.com/docs/rfbproto.pdf>.

[24] Secure Hash Algorithm. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 6. 1. 2008, last modified on 27. 10. 2010 [cit. 2010-11-29]. Dostupné z WWW: <http://cs.wikipedia.org/wiki/Secure_Hash_Algorithm>.

[25] *TeamViewer* [online]. [cit. 2010-11-15]. TeamViewer. Dostupné z WWW: <http://www.teamviewer.com/cs/>.

[26] TripleDES. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 19. 6. 2007, last modified on 23. 9. 2010 [cit. 2010-11-16]. Dostupné z WWW: <http://cs.wikipedia.org/wiki/TripleDES>.

[27] *Ultra VNC* [online]. 2008, 2010 [cit. 2010-11-16]. Remote Support Software. Dostupné z WWW: <http://www.uvnc.com/>.

[28] *VNCSharp* [online]. 14. 11. 2006, 9. 1. 2009 [cit. 2010-12-08]. Architecture Overview. Dostupné z WWW: <http://cdot.senecac.on.ca/projects/vncsharp/architecture.html>.

[29] ZAKHOUR, Sharon. *Java 6 : Výukový kurz*. první vydání. Brno : Computer Press, a.s., 2007. 534 s. ISBN 978-80-251-1575-6.

[30] *Zlib Home Site* [online]. 1996, Last updated November 7th, 2010 [cit. 2010-11-01]. Zlib. Dostupné z WWW: <http://www.zlib.net/>.

Seznam použitých zkratek

AES	Advanced Encryption Standard
API	Application Programming Interface
BC	Bouncy Castle
CFB	Cipher FeedBack
DES	Data Encryption Standard
DMS	Data stream Modification
DNSSEC	Domain Name System Security extensions
FIPS	Federal Information Processing Standard
GCC	Generic Conference Control
GPL	GNU General Public License
IANA	Internet Assigned Numbers Authority
IPsec	Internet Protocol security
ITU	International Telecommunication Union
JAR	Java Archive
JCE	Java Cryptographic Extensions
JDK	Java Development Kit
JRE	Java Runtime Environment
LAN	Local Area Network
MD5	Message-Digest algorithm 5
NAT	Network Address Translation
OTP	One-Time Password

PDU	Protocol Data Unit
PGP	Pretty Good Privacy
PKI	Public Key Infrastructure
RDP	Remode Desktop Protocol
RFB	Remote Framebuffer
RFC	Request For Comments
RMI API	Remote Method Invocation Application Program Interface
SHA	Secure Hash Algorithm
S/MINE	Secure / Multipurpose Internet Mail Extensions
SSH	Secure Shell
SSL	Secure Socket Layer
SSL/TLS	Secure Socket Layer / Transport Layer Security
TCP/IP	Transmission Control Protocol / Internet Protocol
URL	Uniform Resource Locator
VNC	Virtual Network Computing
VoIP	Voice over Internet Protocol
WEP	Wired Equivalent Privacy

Seznam příloh

Příloha A: Obsah přiloženého CD

xfried04.pdf – text diplomové práce ve formátu PDF

Složka s programy: aplikace

Manuál ke spuštění aplikace:
ManualSpusteni.txt

Složka spouštěcích souborů:
SpousteciSoubory\

spuštění aplikace pomocí příkazové řádky:
java -cp bcprov-jdk16-146.jar;jrdesktop.jar jrdesktop.main – systém MS Windows
java -cp bcprov-jdk16-146.jar:jrdesktop.jar jrdesktop.main – systém Linux

Složka zdrojových kódů aplikace:
jrdesktop\