JIHOČESKÁ UNIVERZITA V ČESKÝCH BUDĚJOVICÍCH

Pedagogická fakulta

Katedra informatiky

Digitální steganografie

Diplomová práce

Ing. Ladislav Beránek, CSc. 2009 Bc. Helena Kociánová

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracovala samostatně a že jsem veškerou použitou literaturu uvedla v seznamu použité literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. V platném znění souhlasím se zveřejněním své diplomové práce, a to v nezkrácené podobě pedagogicku fakultou elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách.

V Českých Budějovicích dne

Abstrakt

Digitální steganografie je metoda, která umožňuje ukrývání souborů do jiných, nejčastěji multimediálních souborů (obrázků, hudby, videa). S rozvojem výpočetní techniky došlo k jejímu rozmachu v souvislosti s možným použitím při ochraně autorských práv nebo při skrytém přenosu informací. Velkému zájmu se digitální steganografie těší také proto, že ji lze použít i tam, kde je z nějakého důvodu omezeno použití kryptografie (např. legislativně). Tato práce mapuje problematiku digitální steganografie a zahrnuje aplikaci využívající tuto metodu.

Abstract

Digital steganography is a technique for hiding data mostly into multimedia files (images, audio, video). With the development of information technology this technique has found its use in the field of copyright protection and secret data transfer, could be even applied in places where is limited possibility of using cryptography (e. g. by law). This thesis gives insight into digital steganography and contains an application using this technique.

.

OBSAH

ÚVOD

Digitální steganografie je metoda, která umožňuje ukrývání souborů do jiných, nejčastěji multimediálních souborů (obrázků, hudby, videa). S rozvojem výpočetní techniky došlo k jejímu rozmachu v souvislosti s možným použitím při ochraně autorských práv nebo využitím pro skryté zasílání zpráv. Velkému zájmu se digitální steganografie těší v této oblasti i proto, že ji lze použít k zabezpečení citlivých údajů i tam, kde je z nějakého důvodu omezena možnost použití kryptografie (např. legislativně).

Impulsem ke zpracování tohoto tématu byl článek s názvem „Tajomstvá steganografie“ [1], v němž autor populární formou přibližuje základní informace k dané problematice. Posláním této práce je tedy seznámení se s digitální steganografií.

V teoretické části je zmíněna stručná historie steganografie, zbytek této části je pak věnován především digitální steganografii, s malou odbočkou k informacím o digitálnímu vodoznaku. Je zde popsán princip digitální steganografie, užívané postupy při ukrývání a vyjímání souborů, využití steganografie a její odhalování (tzv. steganalýza). Celou teoretickou část prostupují odkazy na zajímavé studie dostupné na internetu.

V praktické části je pak popsán proces, který vedl k vytvoření jednoduchého steganografického programu pro vložení a následné vyjmutí souboru z jiného souboru.

Během psaní této práce vyvstal problém s nalezením ekvivalentních českých pujmů k anglickým. Proto jsou v místech, kde bylo možné použít vhodný český výraz, uvedeny v závorkách anglické termíny. Tam, kde se vhodný výraz nalézt nepodařilo, byla dána přednost anglickým pojmům před zaváděním nových.

1. DIGITÁLNÍ STEGANOGRAFIE

Pojem ***steganografie*** má svůj původ v řečtině, ve slovech stegos „skrytý“ a grafia „psaní“, čili „skryté psaní“ („cover writing“) [2]. Snaha popsat podstatu steganografie co nevýstižněji přinesla celou řadu více méně podobných definic, z nichž za všechny lze vybrat tyto tři:

Steganografie *je metoda skrývání osobních nebo citlivých informací v něčem, co se na první pohled nejeví jako neobvyklé [3].*

Steganografie *je umění a věda skrývání faktu, že jde o komunikaci [4],[5].*

Steganografie *je metoda ochrany dat skrytím souvislosti, ve které jsou přenášena[6].*

Cílem steganografie je tedy ukrýt zprávu (informaci, data) tam, kde by ji nikdo nečekal, a zároveň tak, aby její přítomnost nebyla detekována.

1.1 Historie

Hojné využívání steganografických technik je známo už z dob starého Řecka a Říma. Zatímco v souvislosti se starým Řeckem jsou nejčastěji zmiňovány metody posílání tajných zpráv za pomoci tetování na hlavách otroků (na oholenou hlavu byla zpráva vytetována a poté, co vlasy dorostly, vydal se otrok zprávu doručit) nebo tabulek (s vyrytým textem) zalitých do vosku, Řím je spojován s používáním neviditelného inkoustu, tj. psaní mezi řádky za pomocí látek (např. mléka, octu, ovocných šťáv), které při zahřátí ztmavnou. Později byl vynalezen inkoust chemický (k zviditelnění písma dochází za pomoci chemické reakce), který byl spolu s technikou mikroteček (zmenšených dokumentů a fotografií) použit i za druhé světové války. Moderní obdobou neviditelného inkoustu jsou například ochranné prvky na bankovkách viditelné pod UV zářením.

Za možný počátek nahlížení na steganografii jako na vědní disciplínu lze považovat konec 15. století, kdy Johannes Trithemius, jeden ze zakladatelů moderní kryptografie, publikoval dílo s názvem „Steganographia“. V něm popsal rozsáhlý systém pro ukrývání zpráv v nevinném textu. Ovšem za skutečnou knihu o steganografii

je považována práce ze sedmnáctého století s názvem „Steganographica“, jejíž autorem je Gaspar Schott. [5]

Z moderní historie stojí za to zmínit také jedno prvenství, a to první konferenci na toto téma v roce 1996 [8].

1.2 Moderní steganografie

Moderní doba přeje renesanci starých myšlenek. V současnosti znamená bezpečnost komunikace nejen její utajení, ale i maskování, takže steganografie zaznamenává svůj rozmach v síťové komunikaci nebo při ochraně citlivých údajů a autorských práv.

Digitální steganografie se od té klasické liší snad pouze v tom, že v ní lze teoreticky ukrývat cokoliv, co lze napsat v bitové podobě (bitstream). V praxi to samozřejmě není tak jednoduché. Nejběžnějším způsobem užití je ukrývání jednoho souboru do jiného, nejčastěji obrázku, hudby, nebo videa. Je to dáno velkým množstvím těchto souborů na internetu, které tak může sloužit jako dobré „křoví“ pro skrytou komunikaci. S masivním nárůstem užíváním internetu souvisí i další uplatnění steganografických myšlenek, a to v oblasti ochrany autorských práv za pomoci digitálního vodoznaku (digital watermarking). (viz kap. 1.4)

1.2.1 Steganografický proces

Stegosystém *– systém pro ukrytí a následné vyjmutí zprávy [9]*

Na úvod je zde uvedena rovnice obecného popisu steganografického procesu [6]:

cover medium + embedded message [+stegokey] = stego-medium

Krycí médium (cover medium) je soubor, do kterého ukrýváme zprávu ***(embedded message)***. Někdy se v procesu ukrývání používá i takzvaný ***stego-klíč (stego-key)***, který může sloužit k vyššímu zabezpečení a jeho znalost je nutná pro proces extrakce souboru z krycího média. Výsledkem procesu ukrytí zprávy je ***stego-médium***. V literatuře se lze často setkat i s jiným označením:

- *cover medium (carrier medium, covertext)*
- *embedded message (hidden message, embedded data)*

1.2.2 Typy steganografie [5]

Obecně se rozlišují tři typy steganografie:

- jednoduchá steganografie (Pure Steganography),
- steganografie s privátním klíčem (Private Key Steganography)
- steganografie s veřejným klíčem (Public Key Steganography)

Jednoduchý steganografický systém nevyžaduje žádnou směnu kódu (např. stego-klíče). Jde o nejméně zabezpečený přístup, který spoléhá na to, že kromě odesílatele a příjemce o ukryté zprávě nikdo jiný neví.

Steganografie s privátním klíčem vyžaduje výměnu tohoto klíče mezi komunikujícími stranami, protože je nedílnou součástí dekódovacího procesu. Proto je tento typ steganografie náchylnější k zachycení probíhající komunikace.

Steganografie s použitím veřejného klíče je obdobou kryptografie s veřejným klíčem (Public Key Cryptography). Pro zabezpečení komunikace využívá veřejný klíč k ukrytí zprávy a na druhé straně soukromý klíč (matematicky svázaný s veřejným) k vyjmutí ukrytých dat.

1.2.3 Metody

Nejjednodušší programy pracují na principu vložení ukrývané informace na konec souboru (tzv. ***injection***).[3],[1] To vede zákonitě k navýšení velikosti souboru o velikost ukrývané zprávy. Většina dalších metod digitální steganografie je založena na vyhledávání tzv. ***redundantních bitů*** v krycím médiu a jejich nahrazení bity ukrývané zprávy. K tomuto účelu se obvykle používají volné bajty, resp. vhodná pole, v hlavičkách souborů, nebo se při hledání volných bitů (speciálně u multimediálních souborů) využívá nedokonalosti lidského vnímání, v případě obrázků například neschopnosti rozeznat blízké odstíny barev, v případě hudebních souborů pak vlastností lidského sluchu, který obvykle dokáže vnímat rozsah jen 20 Hz – 20 kHz.

Metody lze rozdělit do tří hlavních skupin:

LSB (Least Significant Bit), maskování a filtrování, transformační techniky [10]

Metoda ***LSB*** spočívá v nahrazení nejméně významného bitu, příp. dvou nejméně významných bitů, v bajtu jedním, resp. dvěma, bity ukrývané zprávy. Nabízejí se dvě

možnosti jak to provést, buď přepsáním hodnoty bitu, nebo přičtením (odečtením) bitu [11]. Z teorie pravděpodobnosti vyplývá, že teoreticky je při této metodě nutná změna pouze poloviny potřebných bitů.

Metody ***maskování*** a ***filtrování*** jsou založeny na využití nedokonalosti lidského vnímání. S jejich pomocí lze ještě navýšit možnou ***kapacitu pro ukrytí zprávy (hiding capacity)*** nebo zvýšit ***robustnost*** stego-media [12]. (viz kap. 1.2.4)

Mezi ***transformační techniky*** patří využití diskrétní kosinové transformace (DCT), diskrétní fourierovy transformace (DFT) a vlnkové transformace (wavelet transformation) [12]. Jsou vytvořeny tak, aby odolávaly, nebo naopak využívaly metod populárních komprimačních algoritmů [12, 10].

Jako krycí médium lze použít cokoliv, co obsahuje nadbytečné, nebo nepodstatné informace (redundantní bity). Podporováno je velké množství nejrůznějších formátů.

Obrázky

Jak už bylo zmíněno výše, jsou obrázky (grafické soubory) velmi populárním médiem (cover mediem) pro ukrývání informací. Vhodné jsou jak pro svůj hojný výskyt na internetu, tak pro svou „úschovnou“ kapacitu.

Nejběžněji uváděným příkladem steganografie na obrázcích je ukrytí textu (v tomto případě písmene A) do tří pixelů obrázku ve formátu BMP pomocí metody využití nejméně významného bitu LSB (Least Significant Bit).

Máme 24-bitový obrázek v rozlišení 1024x768 pixelů. Jeho celková velikost je přibližně 2,25 MB. Z metody LSB víme, že můžeme měnit jeden bit v každé barevné komponentě pixelu, čili 3 bity na pixel. Každá komponenta je definována 8 bity, takže maximální velikost ukrývané zprávy je jedna osmina velikosti média, v našem případě tedy cca 228 kB dat.

Písmeno „A“ (v bitové podobě 01000001) lze tedy přidat do tří pixelů obrázku takto:

00101001 11001011 10110110
01000110 10110010 01100111
00111011 10101110 00110011

V původním řetězci (viz výše) změníme podle potřeby poslední bit v každém bajtu). Výsledek vypadá následovně:

00101000 11001011 10110110
01000110 10110010 01100110
00111010 10101111 00110011

Při výběru konkrétního souboru (obrázku, fotografie) a metody pro ukrytí zprávy hrají roli řada věcí:

- typ souboru
- barevná hloubka
- rozvržení plochy
- velikost
- zamýšlený způsob doručení zprávy příjemci

Typem souboru je míněná především použitá komprese. V případě ztrátové komprese (typicky formát JPG) se dlouho věřilo, že nebude ke steganografii vhodná, protože už malá změna může způsobit viditelný šum. Ovšem s masivním rozšířením a vysokou popularitou tohoto formátu bylo jen otázkou času, než se přijde s metodou, která by umožnila efektivní ukrývání i do souboru tohoto typu.

Proces komprimace jpegu je rozdělen do dvou fází – ztrátové a bezztrátové. V první fázi je z každého bloku 8x8 pixelů získáno pomocí diskrétní kosínové transformace (DCT) 64 koeficientů, které jsou následně kvantifikovány. Po této fázi lze využít LSB kvantifikovaných koeficientů k ukrytí dat. Změna jednoho koeficientu ovlivní všechny pixely daného bloku, takže nedojde k viditelným změnám. [5]

Řada steganografických algoritmů pro formáty se ztrátovou kompresí používá rychlou fourierovu transformaci pro zjištění, kam do souboru by šlo něco uložit, aniž by to bylo viditelné.

Velmi vhodné jsou pro steganografii obrázky s velkou barevnou hloubkou. Ideální je v tomto 24-bitový BMP nebo 8-bitový obrázek ve stupních šedi. Velmi důležité je i rozložení plochy obrázku, resp. vhodný je obrázek s minimem homogenních (jednobarevných) ploch. Ať už nízká barevná hloubka, tak přítomnost

rozsáhlých jednobarevných ploch může způsobit nárůst viditelného šumu po ukrytí zprávy. Zajímavou možností je využití přebytečných bitů převodem obrázku z 256 na 32 barev, čímž se uvolní 3 bity v každém bajtu [1].

V neposlední řadě hraje roli i způsob, jakým chceme skrytou zprávu doručit příjemci, resp. příjemcům. Rozhodneme-li se například předávat své skryté zprávy prostřednictvím jednoho z mnoha milionů portálů, na kterých uživatelé sdílejí své fotografie, mohlo by použití fotek ve formátu BMP působit podezřele.

Další možností je využití tzv. Patchwork algoritmu. Ten spočívá v pseudonáhodném výběru dvojice pixelů, z nichž světlejší udělá ještě světlejším a tmavší pixel ještě tmavším. Tato malá změna není postřehnutelná, ale změna kontrastu poslouží jako vzor pro ukrytou zprávu [8]. Tohoto algoritmu lze použít i u audio souborů pro zvýšení kontrastu amplitudy párů náhodně zvolených vzorků zvuku v celém zvukovém souboru. Filtrováním je pak odstraněn vysokofrekvenční šum vzniklý během procesu [8].

Kromě zmíněných metod je zde ještě možnost generování fraktálních obrázků nad ukrývanými daty (zprávami). Jejich výhodou je nepochybně to, že odpadá potřeba výběru vhodného krycího media [1].

Text

Ukrývat informace v textu lze buď na úrovni sémantiky, formátování, nebo syntaxe. V prvním případě hrozí reálné nebezpečí nevhodného výběru slov a synonym, které mohou svou nepřirozeností vzbudit podezření na přítomnost skrytého obsahu. Pokud jde o formátování, nabízí se řada možností:

- přidání prostoru mezi znaky
- přidání mezer a tabulátorů na konce řádků (toto lze snadno odhalit v textových editorech umožňujících zobrazení neviditelných – pomocných – znaků)
- posun mezer mezi řádky nebo slovy

Zásah do syntaxe jako možnost ukrytí zprávy se týká hlavně zdrojových souborů. Zde se využívá například ignorování bílých mezer kompilátorem, nebo možnosti dvojího zápisu tagů, např. ukrytí řetězců 101100 a 010011 [13]:

Stego key

<tag>, </tag>, <tag/> -> 0

<tag_>, </tag_>, <tag_/> ->1

Stego data

<user_><name>Alice</name_><id_>01</id></user>

<user><name_>Bob</name><id>02</id_></user_>

Pozn.: Pro větší přehlednost je v příkladu použit znak „_“ místo znaku mezery.

Zvuk [5]

Zvukové soubory jsou dostatečně dlouhé pro ukrytí malého množství informace. Kromě velmi oblíbeného formátu MP3 existují steganografické programy s podporou dalších formátů, např. WAV, PCM, MIDI. Při převodu zvuku do digitální podoby dochází k jeho vzorkování a kódování.

Z hlediska digitálního zvuku ve steganografii hrají roli vzorkovací frekvence, počet kanálů a také rozmanitost zvukové stopy. U hudebního souboru s nízkou hodnotou těchto veličin způsobí vložení skryté zprávy slyšitelný šum.

Low bit encoding je obdobou LSB u obrázků. Dochází při něm k nahrazení LSB každého vzorkovacího bodu kódovaným binárním řetězcem. Při kompresi zvuku jsou kódovány jen části, které je člověk schopen vnímat, což snižuje účinnost této metody.

Fázové kódovaní (phase coding) je tam, kde jej lze použít, prokazatelně nejvíce efektivní metodou z hlediska poměru signálu k šumu. Postup této metody je poměrně složitý, jeho podrobnější popis je k nalezení ve studii [5].

Další metodou skrývání dat ve zvukový souborech je metoda zvaná ***spread spektrum***. Spočívá v „rozprostření kódovaných dat skrz maximální rozsah frekvencí“, což se ve výsledku jeví jako náhodný šum [14].

Poslední metodou je ***echo hiding***. Při ní se v krycím mediu provádějí změny charakteristické pro různá prostředí místo vytváření náhodného šumu.

Jinou možností je pak ukrytí zprávy je ***vložení extra zvuku*** [3].

Video [3]

Při ukrývání informací do videa se s využitím diskrétní kosinové transformace (DCT) provádí změna každého snímku videa jen do té míry, aby výsledný dojem nebyl ovlivněn. Dochází k náhradě hodnot určitých částí obrázku, obvykle jejich zaokrouhlení.

Disk, přenosná média

V případě aplikace steganografických metod na disky, příp. přenosná zařízení, se dají využít dva hlavní přístupy:

- alokace nevyužitého místa v sektorech (prostoru mezi koncem legitimního souboru a koncem sektoru) [1]
- alokace prostorů na viditelném disku, nebo místa, které nebylo disky zatím akokováno [1]

Existuje ***steganografický souborový systém***. Úkolem takového systému je umět pracovat jak s viditelnými, tak i ukrytými soubory, dokázat zkreslit údaje o počtech souborů a mazat oba typy souborů podobně, aby se skrytá sekce nepřepisovala chaoticky. K ukrytým datům lze obvykle přistupovat pomocí steganografického programu, nebo za pomocí trojského koně.

Asi nejznámějším steganografickým systémem je StegFS v souborovém systému ext2fs pro linux [6].

Síťové protokoly

Ke skryté komunikaci (přenášení dat) lze použít celou řadu protokolů, např. IP, UDP, TCP, v případe zvukového přenosu transportní protokol RTP a doplňkový RTCP [15]. Využívá se při tom některých polí hlavičky, která jsou během přenosu měněna, nebo nahrazována s menší pravděpodobností. O této problematice je možné se více dočíst ve studii [5] a studii [15] zaměřené na steganografii ne VoIP.

1.2.4 Steganografické algoritmy

K hodnocení steganografického algoritmu se nejčastěji používají tato kritéria: kapacita „úložného“ prostoru (hiding kapacity), vjemová stálost (perceptual transparency***)*** a robustnost (robustness).

Kapacitou „úložného“ prostoru (hiding capacity) je myšlena poměrná velikost ukrývané zprávy k velikosti krycího média. Ve studii [11] je diskutován jeden z možných přesnějších přístupů k této veličině.

Vjemová stálost (perceptual transparency) vyjadřuje míru vjemové podobnosti krycího média se stego-médiem. Je snahou (především pak při použití digitálního vodoznaku) co možná nejméně ovlivnit vlastnosti krycího média, aby nedocházelo ukrytím informace k viditelnému, resp. slyšitelnému, nárůstu šumu.

Robustnost (robustness) obnáší schopnost ukryté zprávy odolávat nejrůznějším zásahům, například transformacím, zaostřování či rozmazání, škálování a rotaci, ořezání, ztrátové kompresi, převodu z digitálního na analogový signál a zpět [12]. Ve studii [16] je robustnost definována jako „kvantifikace dekódovací spolehlivosti v přítomnosti kanálového šumu“.

Dalšími kriterii hodnocení steganografického algoritmu mohou být *detekovatelnost (detectability),* která hodnotí schopnost detekce skryté zprávy [16], nebo *odolnost proti falšování (Temper resistance)* kvantifikující obtížnost nahrazení, nebo padělání zprávy ve stego-médiu [12].

Jednou z možností, jak odolat aspoň některým změnám, například ztrátové kompresi, je ukrývat informaci do významných částí souboru.

1.2.5 Steganalýza

„Steganalýza *je postup napadení steganografické metody detekcí, destrukcí, vyjmutím (extrakcí) nebo pozměněním ukrytých dat.“ [12]*

Na stegoanalýzu je možné nahlížet ze tří hledisek:

- množství informací, které jsou k dispozici
- způsob detekce
- cíl steganalýzy

Z hlediska množství informací, které jsou k dispozici, lze stegoanalýzu rozdělit na:

- Stego – only
- Known – message
- Chosen – steganography
- Chosen – message
- Known – cover
- Known – steganography

Stego – only znamená, že při steganalýze je známo pouze stego-médium. Jde o nejslabší formu útoku proti steganografické metodě, ovšem pokud je cílem zjistit pouze to, zda soubor obsahuje ukrytá data a nikoliv už jejich obsah, může být dostačující.

V případě ***know – message*** útoku je k dispozici pouze ukrytá zpráva (embedded message).

Při ***chosen – steganography*** je k dispozici jak stego-médium, tak použitý algoritmus, tedy software, kterým byla zpráva ukryta. Podobně je tomu u ***chosen – message*** útoku, kdy společně s algoritmem je místo stego-média známa ukrytá zpráva.

Know – cover je označení pro útok, při kterém je k dispozici stego-médium a krycí médium. Toho lze dobře využít při vyhodnocování změn v krycím médiu způsobených ukrytím dat, která lze následné zobecnit pro statistickou steganalýzu.

Nejvíce informací zahrnuje kategorie ***known – steganography***, kdy je znám jak použitý algoritmus, tak krycí médium a ukrytá zpráva. [17], [12]

Z hlediska způsobu detekce lze steganografické metody rozdělit na:

- vizuální a poslechovou analýzu
- strukturální analýzu
- statistická analýzu

Při analýze pomocí zraku nebo sluchu je důležitá citlivost lidského vnímání. Změna v krycím médiu může vést k drobným odchylkám, které mohou vyvolat pocit (dojem), že něco není po vizuální, nebo zvukové stránce v pořádku. Ve steganografii aplikované na text může podezření vyvolat i vnímání textu (gramatika, sémantika).

Strukturální analýza se soustředí na odchylky ve formátování, například nadbytek řádků a bílých mezer bez zřejmé logiky.

Statistická analýza pak vychází z anomálií vyskytujících se ve stego-médiu, např. v paletových obrázcích chaoticky poskládaná paleta, zdvojení barvy v paletě, nebo odchylky v histogramu. Podezření mohou vyvolat i neobvyklé, nebo opakující se vzory bez příčiny, stejně jako přítomnost šumu v obrázcích, příp. hudbě, což je problém především u formátů používajících ztrátovou kompresi. [6]

Pokud je při analýze k dispozici stego-médium i původní krycí médium, je o něco jednodušší vytvořit statistickou metodu, která by umožnila efektivněji určit pravděpodobnost možného výskytu skrytých informací ve stego-médiu vytvořeném konkrétním programem. V případě statistické analýzy bez konkrétního krycího, nebo stego média je potřeba mnoho materiálu, aby bylo možné stanovit, co je pro daný objekt normální. [5]

Řada analytických programů je přímo určena k detekci ukrytých dat konkrétním programem (např. program Stegodetect dokáže najít skryté informace v jpg souborech [6]), nebo se soustředí na konkrétní typ krycího média (např. program StirMark, který je zaměřen na stegoimages, funguje na principu zanášení chyb do obrázku – jako by byl vytisknut a následně neskenován – a sleduje množství šumu) [8].

Existuje i řada metod nezávislých na formátu krycího media. Takové metody měří entropii redundantních bitů. Předpokládá se, že ukrytí zprávy způsobuje vyšší entropii krycího média. [4]

Mnoho analytických programů je vybaveno „hrubou silou“ ke zjištění přítomnosti skryté zprávy, nebo k prolomení hesla, příp. šifry. [8]

Současné detektory umí zjistit nejen přítomnost ukryté zprávy, ale i její délku, dokonce i najít stego-klíč. Proto se propracovanější steganografické programy snaží udržet stego-médium co nejpodobnější krycímu médiu, aby tak unikly pozornosti statistické analýzy. Detekci stěžuje jak použití šifrování, tak randomizace ukrývaného obsahu. [11], [17]

1.2.6 Využití

Jak už bylo několikrát naznačeno v předcházejícím textu, má steganografie široké použití. V prvé řadě ji lze použít k ochraně citlivých dat (např. ve firmách jako doplňkovou metoda ochrany před průmyslovou špionáží) speciálně tam, kde nelze volně

užít kryptografii (viz kap. 1.3). Druhou významnou oblastí je její komerční využití k ochraně autorských práv. V této souvislosti se mluví o pojmech watermark (vodoznak) a fingerprint (otisk). Rozdíl mezi nimi spočívá v tom, že u vodotisku je použita stejná značka pro všechny objekty, kdežto u otisku má každý objekt svou specifickou značku [10] (více v kap. 1.4). K dalším aplikacím steganografie patří např. vkládání klíčových slov do obrázků a jiných souborů, která usnadňují práci vyhledávacím strojům, dále také časové značky ve videu pro synchronizaci se zvukem nebo využití v tzv. pay-per-view aplikacích [12]. Některá média se v minulosti zmínila o možnosti zneužití steganografie teroristy při plánování své činnosti. S odvoláním na studii [3] jde spíše o spekulace.

1.2.7 Destrukce ukryté zprávy

Zničit ukrytá data jde mnoha způsoby. V případě metody LSB stačí pouhé vynulování těchto nejméně významných bitů, nebo převod do formátu se ztrátovou kompresí (př. z BMP na JPG). Na obrázky lze také aplikovat nejrůznější transformace (rotace, ořezání, škálování a jiné). Využití kombinace transformací je účinnější než využití pouze jedné. Při použití textové steganografie lze skrytou zprávu zlikvidovat například smazáním nadbytečných mezer a tabelátorů.

1.3 Steganografie vs. kryptografie

Steganografie je často zmiňována v souvislosti s kryptografií. Zatímco snahou kryptografie je naložit se zprávou tak, aby její obsah byl „čitelný“ pouze za splnění určitých podmínek, jako je např. znalosti šifrovacího klíče, snahou steganografie je skrýt samotnou existenci této zprávy, takže po jejím nalezení ji lze normálně přečíst. Použití steganografie může být alternativou použití kryptografie na místech, kde je možnost kryptografie z nějakého důvodu (např. legislativně) omezena. Obě přístupy mají své silné i slabé stránky. K dosažení co nejvyšší úrovně zabezpečení je tudíž vhodné tyto techniky kombinovat.

1.4 Digitální vodoznak (digital watermark)

Existují tři typy digitálních vodoznaků: viditelný, neviditelný robustní a neviditelný křehký. Viditelný vodoznak umožňuje použití média, ale zároveň ukazuje, komu dotyčné médium patří, případně, kde je možné získat o médiu více informací. Tento typ vodoznaku je v podstatě atributem krycího média, takže ho nelze považovat za steganografii [10].

Neviditelný robustní vodoznak slouží k detekci zneužití média. Měl by být navržen tak, aby dokázal odolávat případným manipulacím s médiem, tedy různým transformacím. Odolnost vodoznaku spočívá v tom, že jej nelze z média vyjmout, aniž by nedošlo k nepřípustné degradaci média.

Neviditelný křehký vodoznak se nachází například na snímcích z digitálního fotoaparátu a slouží pro ověření autentičnosti snímku. Zpravodajské agentury ho můžou požadovat jako důkaz, že snímek nebyl modifikován.[5]

Kromě vodoznaku je třeba mít k dispozici kodér, kterým se vodoznak do média vloží, dekodér pro vyjmutí z média a případně komparátor pro verifikaci. Při verifikaci bývá často potřeba porovnání s originálním (tedy krycím) mediem pro přečtení vodoznaku. Hezkým příkladem může být použití vodoznaku ve zdrojovém kódu. K jeho „vložení“ se využívá faktu, že pořadí některých řádků kódu lze zaměnit, aniž by to mělo vliv na bezchybnou kompilaci. Následné získání vodoznaku pak probíhá právě komparací s původním zdrojovým kódem.

1.5 Dostupnost materiálů na internetu

1.5.1 Studie

Problematika digitální steganografie je velmi rozsáhlá. Na internetu lze sehnat spoustu odborných prací na toto téma, jak obecně pojatých studií, tak v současné době úzce zaměřených na konkrétní oblast využití.

Na řadu studií je odkazováno přímo v textu teoretické části. V následujících odstavcích budou vybrány některé další tituly a případně odkaz dle zaměření dané studie. Položky v seznamu literatury s vyšším číslem, než je tu odkazováno, obsahují

řadu informací společných několika studiím. I když z nich není přímo citováno, byly z nich čerpány informace a jsou tedy v seznamu také uvedeny.

Práce s názvem „Steganographic methods“ [18] nabízí analýzu a testy některých steganografických technik aplikovaných na statické obrázky. Je v ní ukázáno, že skrytí velkého množství dat může pozměnit viditelné charakteristiky obrazu, a vyzdvihnuta důležitost komprese ukrývaných dat.

„On The Limits of Steganography“ [19] zkoumá omezení, která se vyskytují v teorii i praxi. Autoři uvádějí tuto metodu do kontrastu s kryptografií a zmiňují se zde i o užívání veřejného klíče.

„An Overview of Steganography“ [2] nabízí další z mnoha vysvětlení pojmu digitální steganografie. Opět se zaměřuje na obrázky, techniky skrývání, výhody a nevýhody použití jednotlivých formátů obrazu.

Nejen obrazem, ale i využitím hudby ve steganografii, se zabývá studie „Digital steganography: seeing the unseen“ [6]. Dále se v ní autoři zabývají principem stegoanalýzy a zmiňují se o některých programech, které se k této problematice vztahují. Podobnými obecněji zaměřenými pracemi na toto téma jsou „Exploring Steganography: Seeing the Unseen“ [10] a „Hide and Seek: An Introduction to Steganography“ [20].

Většina shromážděných studií se zabývá steganografickými technikami aplikovanými na digitální obrázky [21], analýzou steganograficky ukrytých informací [12, 22, 23, 24, 25] a zmiňují se také o využití v oblasti ochrany autorských práv, jako např. studie s názvem „Steganography And Digital Watermarking“ [13].

1.5.2 Steganografický software

Na internetu je dostupných mnoho steganografických programů. Více než polovina je zaměřena na obrázky. Celá řada studií zaměřených na steganalýzu provádí experimenty nad některými z těchto programů. Mezi nejčastěji jmenované patří OutGuess, Jsteg, JPhide, z ryze textových potom SNOW. Jako analytický program je pak často zmiňován StegDetect [4].

2. CÍLE PRÁCE

Cílem této práce je seznámení se s digitální steganografií, principem jejího fungování a využitím. Součástí je také zmapování dané problematiky na internetu (dostupné studie, programy).

Cílem praktické části je potom vytvoření jednoduchého programu, který s využitím poznatků z teoretické části umožní vložení a následné vyjmutí informace z média, a následná analýza takto vytvořených souborů.

Dalším cílem je vzájemné porovnání programů pro digitální steganografii, které bude provedeno v závislosti na dostupnosti podobně zaměřeného software.

3. METODIKA

Jak už bylo řečeno v teoretické části, existuje celá řada programů pro ukrývání dat. Ze získaných informací o problematice vyplývá, že takovéto programy jsou vytvářeny pro konkrétní formáty krycích médií. Z toho vyplývá jedna zásadní nevýhoda. Společně s klesající dostupností daných typů souborů, způsobenou přílivem lepších formátů, se sníží i možnost využití těchto steganografických programů. Proto byla vyslovena myšlenka, zda je možné vytvořit univerzální steganografický algoritmus, který by umožnil vytvoření stego-média z jakéhokoliv souboru bez ohledu na jeho formát.

3.1 Univerzální steganografický systém

Aby systém mohl být univerzální, měl by co nejefektivněji využívat vlastností, které jsou společné pro všechny vstupní prvky. V našem případě by tedy takový systém měl využívat charakteristik společných pro nejrůznější typy souborů. Z tohoto hlediska se přímo nabízí binární zápis. Veškerý obsah počítače je směsicí 1 a 0. Proběhla úvaha nad těmito otázkami:

1. ***Je možné využít podobnosti birárních zápisů krycího média a ukrývaných dat k nalezení místa, ve kterém vložení ukrývaných dat způsobí nejmenší změnu (tedy nalezení vhodného úseku binárního kódu, ve kterém bude třeba provést jen minimum změn)?***

   K zodpovězení této otázky byl zvolen přístup porovnávání souborů bit po bitu. Výstupy tohoto porovnání mohou být tyto:

   a) získaní procentuální rozdílnosti ukrývaného řetězce bitů v každém bodě porovnání s krycím souborem (při porovnávání podle klíče bit-bit by bylo získáno d_{CM} - d_{EM} údajů při porovnání dvou souborů, kde d_{CM} je délka krycího média a d_{EM} délka souboru k ukrytí v bitech). Takové údaje by mohly například posloužit ke zjištění, v jakém poměru se různě rozdílné řetězce v souboru nacházejí, příp. jestli jde o obecný

trend nevázaný pouze na konkrétní soubor, příp. typ souboru, nebo délku obou porovnávaných souborů.

b) získaní údaje o nalezené nejmenší procentuální rozdílnosti ukrývaného řetězce od krycího. S použitím velkého množství souborů by mohlo být možné zjistit, zda např. existuje nějaký významný rozdíl ve výsledcích v rámci různých formátů, nebo mezi velikostí ukrývaného řetězce a krycího média.

2. Bylo by možné zlepšit výsledky tohoto porovnání volbou hledání podle jiného klíče než 1:1 (bit po bitu)?

Bit po bitu není jedinou možností, jak řetězce porovnat. Nabízejí se i další klíče, např. definovaní 1 (příp. 2 a více) bitu ukrývané zprávy vyšším počtem bitů krycího média. To nebude mít na dosažení lepších výsledků pravděpodobně významný vliv, ovšem rozložení změn po více bitech by teoreticky mohlo zmenšit dopady na krycí médium.

3. Kolik a jaké údaje by bylo nutné uchovat, aby bylo možné ukrytou informaci ze stego-média zase vyextrahovat?

Pokud by se při vytváření programu pro ukrývání informací vycházelo z principu hledání nejvhodnějšího úseku v krycím médiu, zcela nepochybně budou výsledkem procesu ukrývaní údaje, které budou potřebné k opětovnému vyjmutí zprávy. Důležitými informacemi by v tomto případě jistě byla pozice, na které ukrytý soubor v krycím médiu začíná, a délka ukrytého souboru. V případě, že je při ukrývaní dat možné vybrat metodu, kterou se vkládání provede (v případě předchozího bodu postup nalezení vhodného řetězce), určitě je třeba tento údaj přiložit. Aby byla rekonstrukce kompletní, mohla by být další informací koncovka (tedy typ) ukrytého souboru.

4. Jakým způsobem tyto údaje dostat k příjemci?

Pokud tedy existují unikátní údaje potřebné k rekonstrukci (vyjmutí) tajné zprávy, je třeba jejich doručení příjemci. Bylo by vhodné, aby byly součástí stego-média. Nabízí se několik míst, kam tyto údaje „uschovat“:

a) *přidání v podobě bajtů na konec stego-média*

   Vcelku snadný způsob, ale i když půjde řádově o jednotky bajtů, dojde k nárůstu velikosti souboru a zvýšení rizika možného odhalení nadbytečných bajtů některým z analytických programů.

b) *přidání do hlavičky souboru*

   Tato možnost dává smysl jen do okamžiku, než se vrátíme k myšlence univerzálního systému, ve kterém nelze obecně jednoznačně určit místo, do kterého by se tento údaj dal vložit)

c) *přidání do názvu souboru*

   To se může zdát jako nejméně kreativní nápad, ovšem zcela určitě univerzální. Vzhledem k výskytu různých generovaných názvů souborů na internetu by nejspíš takový název lehko zapadl. Může být namítnuto, že pokud někdo název změní, budou údaje potřebné k vyjmutí zprávy nenávratně ztraceny. Ne jen, že jakoukoliv transformací stego-média přijdeme nejspíš o samotnou zprávu, je dost pravděpodobné, že se změní index jejího umístění. Jde o velmi křehký typ digitální steganografie. Účelem však není vytvořit robustní systém, ale vyzkoušet nové možnosti steganografie.

5. Jak velké soubory lze při tomto přístupu vlastně použít?

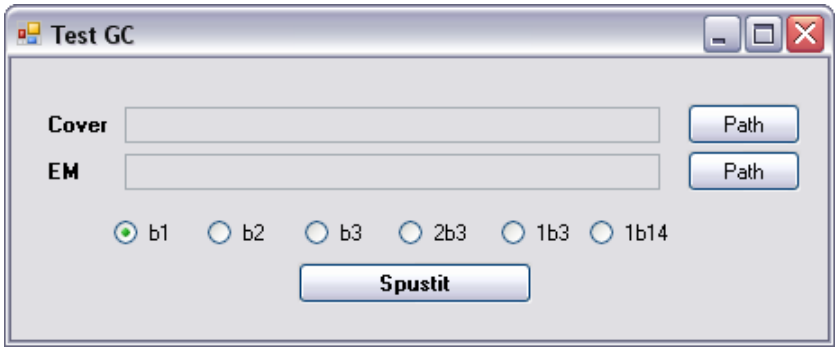
Porovnávání řetězců bit po bitu je jistě náročná činnost. Ideální by bylo mít k dispozici velmi výkonné PC a optimalizovaný kód. Je ale jasné, že obecně bude možné použít krycí soubory o velikosti řádově stovky kB (možná jednotky MB), v případě ukrývaných dat spíše do stovek bajtů (možná jednotek kB).

6. ***Je možné obecně stanovit bezpečný prostor v krycích souborech, ve kterém by vložení skrytých dat nezpůsobilo totální poškození?***

   Určení univerzálního „bezpečného“ prostoru v souboru pro ukrytí zpráv je problém, protože struktura různých typů souborů není stejná a místa, která jsou nutná pro fungování souboru nejsou soustředěna pouze na začátek, příp. na konci souboru. Obecně lze tedy omezit ukládání do části vzdálené bajty od obou krajů souboru.

3.2 Postup

Na základě těchto úvah byl vytvořen jednoduchý analytický program pro porovnávání binárních řetězců. S jeho pomocí byly nasbírány údaje, které byly následně zpracovány. Na základě výsledků byl potom vytvořen jednoduchý steganografický program pro ukrývání a vyjímání zpráv.



Obr. 1: Jednoduchý program pro porovnávání binárních řetězců.

3.2.1 Metody porovnávání

Byly zvoleny tyto způsoby porovnávání řetězců:

b1 – porovnání řetězců 1:1 (tedy 1 bit ukrývané zprávy = 1 bit krycího média)

b2 – porovnávání řetězců 1:2 (1 bit ukrývané zprávy = kombinace 2 bitů krycího média)

b3 – porovnání řetězců 1:3 (1 bit ukrývané zprávy = kombinace 3 bitů krycího média)

2b3 – porovnání řetězců 2:3 (kombinace 2 bitů ukrývané zprávy = kombinaci 3 bitů krycího média)

Doplňková porovnání:

1b3 – porovnání řetězců 1:1 (1 bit ukrývané zprávy = každý 3. bit krycího média)

1b14 – porovnání řetězců 1:1 (1 bit ukrývané zprávy = každý 14. bit krycího média)

V případě porovnávání kombinací několika bitů k jednomu ***(b2, b3)***, příp. dvěma ***(2b3)***, byly bity definovány dvojicí (trojicí) bitů a jejich doplňku, aby bylo nutné vždy znát hodnotu všech bitů ve skupině a nebylo možné vycházet jen z jednoho bitu skupiny (čili zaměnitelnost např. ***b3*** za ***1b3***). V rámci pokusu byla pro označení krycího média použita zkratka CM (cover medium), pro ukrývaná data EM (embedded message).

CM	EM (b2)
00	0
01	1
10	1
11	0

CM	EM (b3)	EM (2b3)
000	0	00
001	1	11
010	0	10
011	1	01
100	1	01
101	0	10
110	1	11
111	0	00

Tab č. 1(a,b): Kombinace bitů v metodách porovnávání řetězců.

3.2.2 Nastavení experimentu

Pro pokus byly jako krycí média náhodně vybrány soubory několika typů (doc, jpg, kar, pdf, ppt, txt/log, xls, mp3). Kritériem byla pouze velikost souboru, která vzhledem k vybraným přístupům musela být minimálně 14x větší než velikost ukrývaného souboru. Jako data v roli ukrývaných informací posloužily tři krátké textové soubory a jeden soubor typu .ico. U nich byla kritériem také pouze velikost.

Obsahem ukrývaných textových souborů byly tyto krátké texty:

EM1.txt Toto je krátká zpráva.

EM2.txt Testovací vzorek 2. Short message.

EM3.txt EM3 webová adresa: http://www.google.cz

Čtvrtým ukrývaným souborem byla ikona programu Task Classifier.

Protože výsledkem mělo být nalezení nejpodobnější části řetězce, bylo z důvodu ušetření času nastaveno, že v případě překročení počtu rozdílných bitů polovinu délky řetězce ukrývané zprávy bude aktuální cyklus ukončen a postoupeno na další pozici v krycím řetězci. Na obsah krycích médií nebyl brán ohled.

Program byl navržen tak, aby pracoval nad složkami, a nebylo tedy nutné spouštět proces vyhodnocování pro každou dvojici souborů zvlášť.

3.3 Způsob vyhodnocení získaných dat

Výstupem tohoto programu byly dva soubory csv. V prvním se ukládaly údaje o vstupních souborech: název, typ (koncovka) a velikost souboru v bajtech. V druhém se pak ukládaly výsledky. Na každém řádku údaje o řetězci s nejmenším počtem rozdílů v bitech takto: název krycího média, název ukrývaného souboru, index počátku řetězce v krycím médiu, počet chybných bitů, procentuální vyjádření počtu neshodných bitů vzhledem k délce ukrývaného souboru a k délce krycího média (viz tabulka).

```
------ Cover files ------
CM1.doc;.doc;50688
CM2.doc;.doc;237056
CM3.doc;.doc;64512
CM4.doc;.doc;78336
CM5.doc;.doc;194048
------ Embedded files ------
EM1.txt;.txt;22
```

Obr. 2: Ukázka výpisu údajů o souborech .

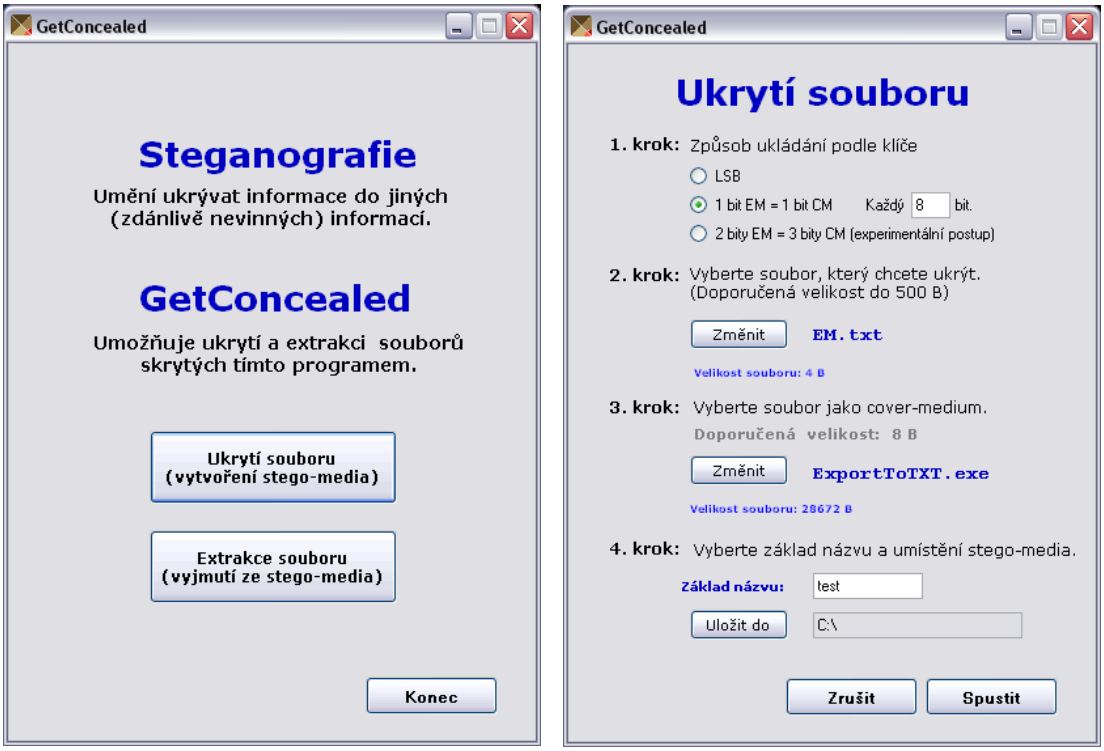
```
CM1.doc;EM1.txt;334838;56;31,82;0,0138
CM2.doc;EM1.txt;364189;57;32,39;0,003
CM3.doc;EM1.txt;455206;54;30,68;0,0105
CM4.doc;EM1.txt;466502;56;31,82;0,0089
CM5.doc;EM1.txt;1356534;54;30,68;0,0035
```

Obr. 3: Ukázka výpisu výstupu porovnávání.

Získané údaje byly zpracovány v MS Excelu. Z výsledků vyhodnocení pak vzešly dvě metody, které byly implementovány v programu na ukrývání a vyjímání zpráv s názvem GetConcealed.

3.4 Program GetConcealed

Jde o experimentální univerzální steganografický program, který ukrývá a vyjímá zprávu bez ohledu na použitý formát krycího média. Kromě implementace dvou vybraných metod, vzešlých z analýzy výsledků pokusu, je jeho součástí i klasická metoda LSB.



Obr č.4: Náhled hlavního okna programu GetConcealed a okna pro ukrytí zprávy.

Na začátku je třeba si vybrat činnost, kterou budeme provádět (tedy ukrytí, nebo extrakci ukrytých dat do/ze souboru).

Ukrývání dat zahrnuje čtyři kroky. V prvním kroku je třeba si vybrat metodu, která bude pro ukrytí použita. V dalším kroku se vybere soubor pro ukrytí. Po jeho výběru se v třetím kroku objeví údaj o doporučené délce krycího média vzhledem k velikosti ukrývaného souboru. Po vybrání krycího souboru se zobrazí buď jeho velikost, nebo upozornění, že soubor není dostatečně veliký. Následuje možnost vybrat si počátek názvu vzniklého stego-média (zbytek názvu budou tvořit údaje o ukrytých datech) a místo, kam bude uloženo.

Po stisknutí tlačítka spustit se provede samotné ukrytí. To probíhá ve dvou fázích (v případě LSB pouze 2. fází):

1. nalezení vhodné části kódu k ukrytí zprávy

2. vložení zprávy (přepis rozdílných bitů)

Výsledkem je stego-médium, jehož název je klíčem k případnému procesu vyjmutí zprávy. Struktura názvu byla zvolena takto:

zvolený_název-metoda[krok]-index_počátku_EM-délka_EM-koncovka_EM.koncovka_CM
(např.: test-2b3-i284571-v23-txt.jpg)

Vyjmutí zprávy probíhá jednoduše vybráním stego-média, ze kterého chceme zprávu vyextrahovat, a místa, kam tuto zprávu potom uložíme.

Obr č. 5: Náhled okna pro extrakci zprávy.

3.4.1 Implementace porovnání a uložení

Oba soubory (krycí médium i ukrývaná zpráva) jsou převedeny přes pole bajtů na pole bitů (BitArray). Aby bylo zachováno pořadí bitů v souboru, je nutné v mezikroku převrátit pole bajtů. Výsledkem je tedy pole bitů souboru, kde počáteční index pole reprezentuje poslední bit souboru. Z toho vyplývá, že porovnávání souborů probíhá od jejích konců. Je však předpoklad, že na výsledek to nemá vliv, a navíc se tím ušetří několikeré převracení pole. Po vytvoření bitových polí přichází na řadu samotné porovnání obou polí podle zvoleného klíče. V případě počtu chybných bitů vyšších než polovina velikosti bitového pole ukrývané zprávy, je cyklus opuštěn a postupuje se o na následující pozici v bitovém poli krycího média. To se provádí, dokud není délka pole krycího řetězce menší než délka ukrývaného řetězce. Poté je index bitu krycího média, kde začalo porovnávání s nejlepším výsledkem (tedy nejnižší hodnotou rozdílných bitů obou polí), použit jako výchozí bod k vložení bitů ukrývané zprávy, které probíhá tak, že v místech rozdílů dojde k nahrazení bitů krycího média bity ukrývané zprávy.

Následně je pole bitů převedeno na bajty a z nich vytvořen nový soubor (stego-médium), do jehož názvu se zaznamenají údaje o ukryté zprávě.

Ukázka klíčové části přepisování bitů při vkládání::

```
// ++++++++ 3 BITY = 2 BITY ++++++++++++
// ++++++++++++++++++++++++++++++++++
 if (rB_2b3.Checked)
 {
     for (int iEM = 0; iEM < EM_pole.Length; )
    {

         if (!EM_pole.Get(iEM) && !EM_pole.Get(iEM + 1))
         {
             if ((CM_pole.Get(jCM) && CM_pole.Get(jCM + 1) &&
                  CM_pole.Get(jCM + 2)) ||
                  (!CM_pole.Get(jCM) && !CM_pole.Get(jCM + 1) &&
                  !CM_pole.Get(jCM + 2))) { }
             else
             {
                    if (CM_pole[jCM])
                    {
                        CM_pole[jCM + 1] = true;
                        CM_pole[jCM + 2] = true;
                    }
                    else
                    {
                        CM_pole[jCM + 1] = false;
                        CM_pole[jCM + 2] = false;
                    }
                }
             }
             else if (EM_pole.Get(iEM) && EM_pole.Get(iEM + 1))
             {
                  if ((!CM_pole.Get(jCM) && !CM_pole.Get(jCM + 1) &&
                    CM_pole.Get(jCM + 2)) ||
                    (CM_pole.Get(jCM) && CM_pole.Get(jCM + 1) &&
                    !CM_pole.Get(jCM + 2))) { }
                   else
                    {
                          if (CM_pole[jCM])
                          {
                              CM_pole[jCM + 1] = true;
                              CM_pole[jCM + 2] = false;
                          }
                          else
                          {
                              CM_pole[jCM + 1] = false;
                              CM_pole[jCM + 2] = true;
                          }
                      }
                 }
                else if (EM_pole.Get(iEM) && !EM_pole.Get(iEM + 1))
                 {
```

```
            if ((!CM_pole.Get(jCM) && CM_pole.Get(jCM + 1)
                && !CM_pole.Get(jCM + 2)) ||
                (CM_pole.Get(jCM) && !CM_pole.Get(jCM + 1)
                 && CM_pole.Get(jCM + 2))){}
                 else
                 {
                     if (CM_pole[jCM])
                     {
                         CM_pole[jCM + 1] = false;
                         CM_pole[jCM + 2] = true;
                     }
                     else
                     {
                         CM_pole[jCM + 1] = true;
                         CM_pole[jCM + 2] = false;
                     }
                 }
        }
        else if (!EM_pole.Get(iEM) && EM_pole.Get(iEM + 1))
        {
            if ((!CM_pole.Get(jCM) && CM_pole.Get(jCM + 1)
                && CM_pole.Get(jCM + 2)) ||
                (CM_pole.Get(jCM) && CM_pole.Get(jCM + 1) &&
                 !CM_pole.Get(jCM + 2))){}
                  else
                  {
                      if (CM_pole[jCM])
                      {
                          CM_pole[jCM + 1] = true;
                          CM_pole[jCM + 2] = false;
                      }
                      else
                      {
                          CM_pole[jCM + 1] = true;
                          CM_pole[jCM + 2] = true;
                      }
                  }
              }
              jCM += 3;
              iEM += 2;
        }
}
//++++++++ bit po bitu (kazdy x-ty) ++++++++++++
//+++++++++++++++++++++++++++++++++
else
{
    for (int iEM = 0; iEM < EM_pole.Length; iEM++)
        {
            if (EM_pole.Get(iEM) != CM_pole.Get(jCM))
               CM_pole[jCM] = EM_pole[iEM];

            jCM += BitStep;
        }

}
```

Princip vyjmutí spočívá v získání údajů z názvu stego-média, které jsou použity v procesu extrakce. Jde o typ metody, index počátku, velikost ukryté zprávy (v bajtech) a typ souboru. Po převedení stego-média do bitového pole je pak danou metodou v místě ukryté zprávy (určeným indexem počátečního bitu) vyextrahováno bitové pole o uvedené délce (v bitech), to je následně převedeno na soubor s udanou koncovkou.

Klíčová část kódu při vyjímání zprávy:

```
// ++++++++ 3 BITY = 2 BITY ++++++++++++
 // ++++++++++++++++++++++++++++++++++
 if (splitMethod[0] == "2")
 {
     int maxIndex = iExEM + (delkaExEM*3/2);
     for (int jCM = iExEM,iEx = 0; jCM < (maxIndex); )
     {
             if ((SM_bitArray.Get(jCM) && SM_bitArray.Get(jCM + 1) &&
                  SM_bitArray.Get(jCM + 2)) ||
                  (!SM_bitArray.Get(jCM) && !SM_bitArray.Get(jCM + 1)
                  && !SM_bitArray.Get(jCM + 2)))
                  {
                           ExEM_bitArray[iEx] = false;
                           ExEM_bitArray[iEx + 1] = false;
                   }
               else if ((!SM_bitArray.Get(jCM) &&
                     !SM_bitArray.Get(jCM + 1) &&
                     SM_bitArray.Get(jCM + 2)) ||
                     (SM_bitArray.Get(jCM) &&
                      SM_bitArray.Get(jCM + 1) &&
                      !SM_bitArray.Get(jCM + 2)))
                    {
                        ExEM_bitArray[iEx] = true;
                        ExEM_bitArray[iEx + 1] = true;
                    }
                else if ((!SM_bitArray.Get(jCM) &&
                           SM_bitArray.Get(jCM + 1) &&
                           !SM_bitArray.Get(jCM + 2)) ||
                           (SM_bitArray.Get(jCM) &&
```

```
                !SM_bitArray.Get(jCM + 1) &&
                 SM_bitArray.Get(jCM + 2)))
            {
                ExEM_bitArray[iEx] = true;
                ExEM_bitArray[iEx + 1] = false;
            }
      else
      {
            ExEM_bitArray[iEx] = false;
            ExEM_bitArray[iEx + 1] = true;
      }
            jCM += 3;
            iEx += 2;
      }
}
//++++++++ bit po bitu (kazdy x-ty) ++++++++++++
//++++++++++++++++++++++++++++++++
else if (splitMethod[0] == "1")
{
    BitStep = Int32.Parse(splitMethod[1]);
    int maxIndex = iExEM + (delkaExEM * BitStep);
    for (int iCM = iExEM, iEx = 0; iCM < maxIndex; iEx++)
    {
         ExEM_bitArray[iEx] = SM_bitArray[iCM];
         iCM += BitStep;
     }
}
```

4. VÝSLEDKY

V této části se nacházejí výsledky provedených pokusů a také výstupy programu GetConcealed.

4.1 Výběr vzorků k testování

Jak už bylo uvedeno v kapitole 3.2.2, jako krycí médium byly v pokusu použity soubory několika typů (doc, jpg, kar, pdf, ppt, txt/log, xls, mp3). V první fázi pokusu byly tyto soubory vybrány pouze s ohledem na minimální velikost vzhledem k velikosti ukrývané zprávy. To se ukázalo jako obrovský problém. Například porovnání 15 krycích médií o celkové velikosti cca 28 MB s jedním textovým souborem o velikosti 41 B trvalo přes 8 hodin. Vzhledem k tomu byl ve druhé fázi testování výběr vzorků omezen na pět zástupců každé kategorie krycích médií a to ve velikosti řádově desítek až několik set kB. Podobně byla omezena i velikost ukrývaných zpráv a to v řádu desítek bajtů. Z toho důvodu byly použity pouze textové soubory. Čtvrtý zástupce ukrývaných dat, ikona, jejíž zpracování bylo také velmi náročné, není součástí souhrnného pohledu, tedy ani závěrů provedených v této práci, ovšem její výsledky opravdu stojí za zmínku.

4.2 Výsledky experimentu

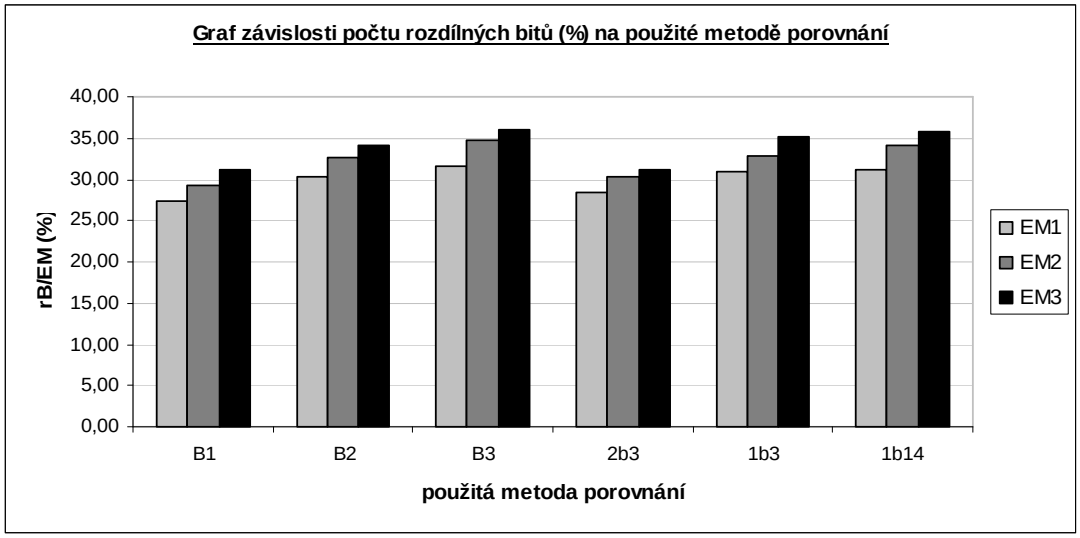
Výsledky z porovnávacího programu byly zpracovány v programu MS Excel. Místo složitých statistických metod, pro které by se hodilo mít jistě mnohem objemnější vzorek dat, byly získané údaje porovnány graficky. Uvažovány byly tyto závislosti:

- závislost počtu rozdílných bitů na použité metodě,
- závislost počtu rozdílných bitů na typu krycího média,

kde počtem rozdílných bitů je myšlen počet rozdílných bitů dělený počet bitů ukrývané zprávy vyjádřený v procentech. Výsledky byly hodnoceny jak pro každý ukrývaný textový soubor zvlášť, tak následně ve společném náhledu.

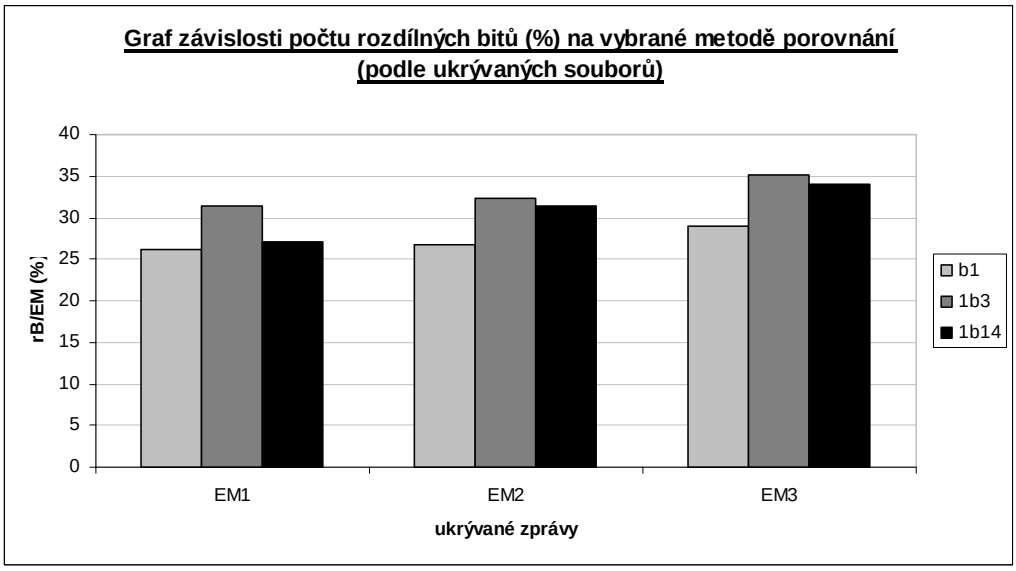
	rB/EM (%)					
	B1	**B2**	**B3**	**2b3**	**1b3**	**1b14**
EM1	27,45	30,34	31,67	28,51	30,97	31,15
EM2	29,16	32,65	34,66	30,38	32,87	34,16
EM3	31,20	34,12	36,09	31,17	35,13	35,72

Tab č. 2: Průměrné hodnoty rozdílnosti pro každou metodu porovnávání.



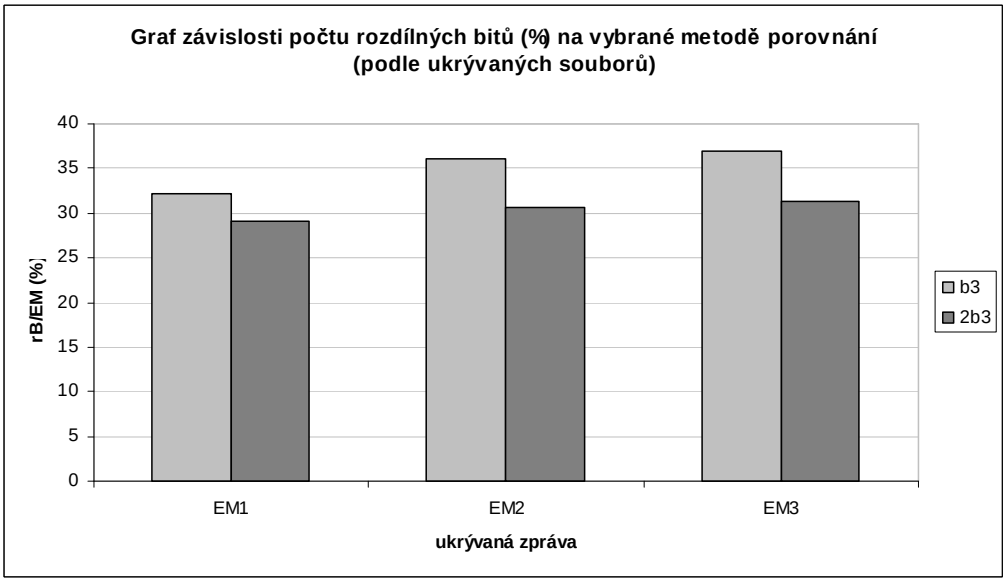
Graf č. 1: Závislost počtu rozdílných bitů (rB) na použité metodě porovnání řetězců.

Ze souhrnných grafů pro všechny ukrývané zprávy na první pohled vidět, že s rostoucí délkou řetězce roste i procento chyb. V porovnání mezi metodami není v některých případech nárůst nárůst této chybovosti až tak výrazný (2b3). Nejlépe si vedla metoda b1, což je klasické porovnání 1:1. Na stejném principu pracují i metody 1b3 a 1b14, které dosáhly horších výsledků. To může být způsobeno menším počtem porovnání. v případě 1b3 3x, v případě 1b14 je to 14x.



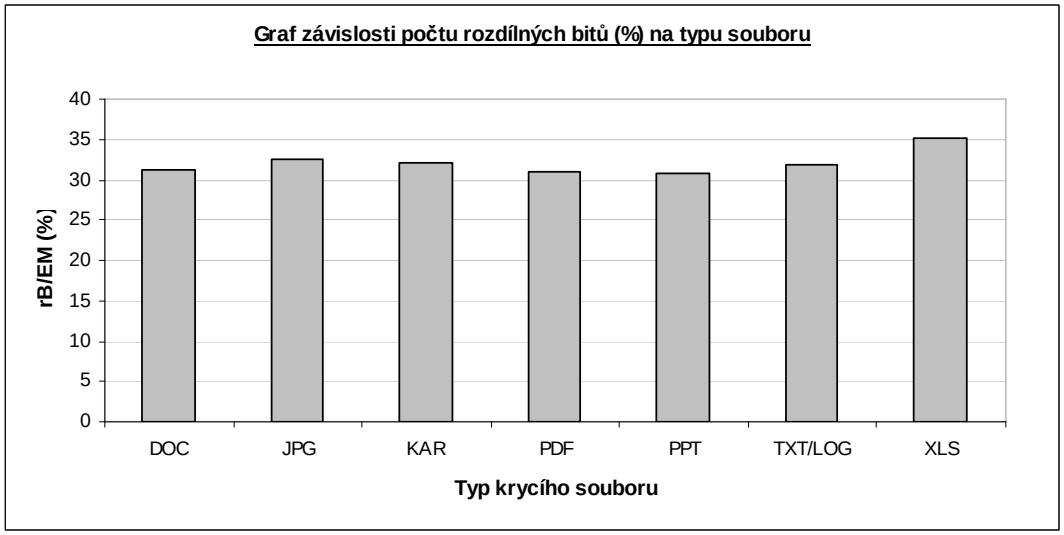
Graf č. 2: Závislost počtu rozdílných bitů (rB) na vybrané metodě porovnání řetězců, řazené podle ukrývané zprávy.

Zajímavé je i srovnání metody b3 a 2b3. V prvním případě jde o porovnání 1 bitu s kombinací 3 bitů krycího média, v druhém pak porovnání kombinace 2 bitů ukrývané zprávy s kombinací 3 bitů krycího média (viz tab č. 1b). Metoda 2b3 vykazuje znatelně lepší výsledky.



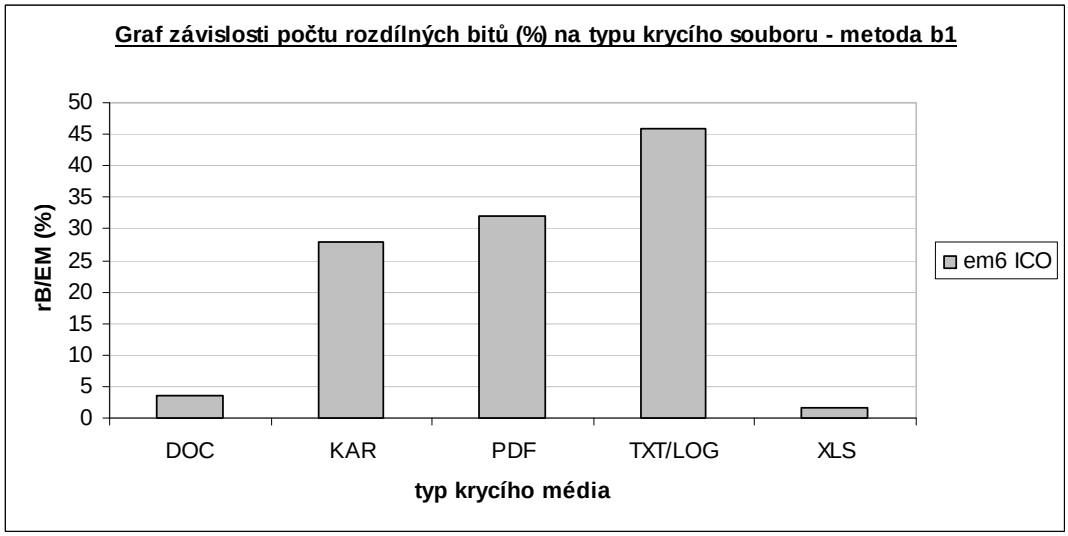
Graf č. 3: Závislost počtu rozdílných bitů (rB) na vybrané metodě porovnání řetězců, řazené podle ukrývané zprávy.

V rámci typů krycích médií pak vycházejí lépe typy souborů s textovým obsahem, nejhůře dopadají porovnání s xls.

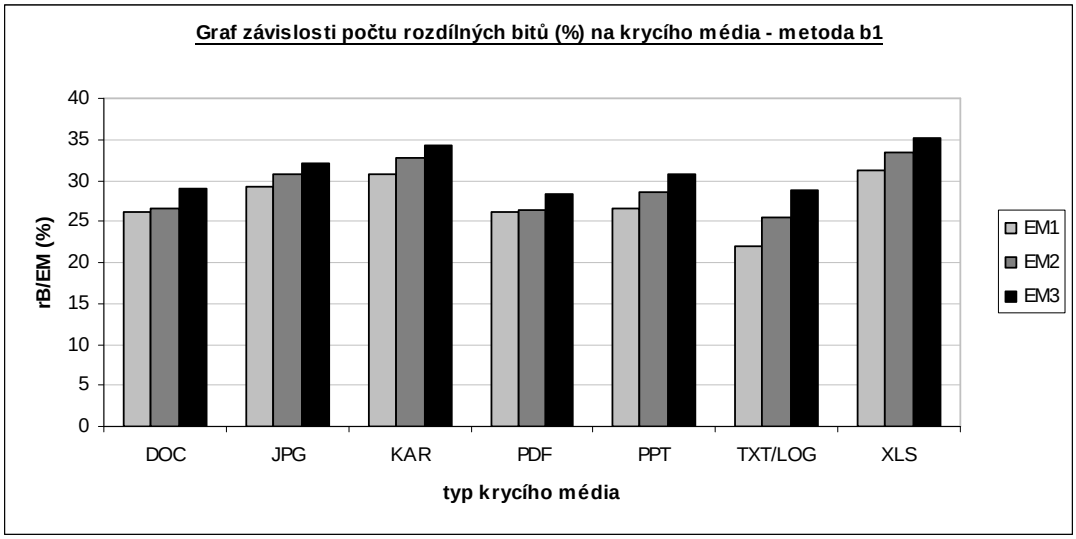


Graf č. 4: Závislost počtu rozdílných bitů (rB) na typu krycího média.

V kontrastu s tím je zde uveden výsledek pro ikonu (jako ukrývanou zprávu) získané kvůli časové náročnosti pouze metodou b1, ve které si xls a doc vedly překvapivě výborně. Může jít o ojedinělý jev, např. způsobený i velikostí porovnávaných souborů.



Graf č. 5: Závislost počtu rozdílných bitů (rB) na typu krycího média – metodou b1.



Graf č. 6: Závislost počtu rozdílných bitů (rB) na typu krycího média – metodou b1.

4.2 Test programu GetConcealed

Z výsledků analýzy dat vyplývá, že porovnání řetězců 1:1 dává nejlepší možný výsledek z použitých přístupů. To nás vlastně vrací k metodě LSB (popsané v teoretické části). Proto byla tato metoda (tedy klasická LSB bez vyhledávání vhodného řetězce) do programu GetConcealed začleněna spolu s metodami bit po bitu (b*x*) s volitelným krokem *x* (upozornění: jde o krok v rámci pole bitů, nikoliv o pozice bitu v bajtu) a metodou 2b3. V rámci prvotních úvah bylo omezeno vkládání na oblast od 51. bajtu od začátku po 3. bajt od konce.

Bylo provedeno několik pokusů o vložení a vyjmutí tajné zprávy s různými možnými výsledky. Předem je dobré zmínit, že metoda LSB a její verze s vyhledáváním a možností nastavení kroku b1 dokáží zprávu jak vložit, tak i vyextrahovat. Metoda 2b3 má bohužel s rekonstrukcí původní zprávy problém. Správně se vyextrahuje pouze něco kolem 50 % bitů, což ale u textu má naprosto destruktivní účinek. Může to být způsobeno chybnou implementací této metody, nebo chybnou úvahou nad jejím fungováním.

Použití na obrázky typu jpg vedlo ke dvěma výsledkům. Prvním jsou různě intenzivně viditelné změny v obrázku, př. Obr č. 7,8 (občas je možné je přehlídnout v pestrých obrázcích, Obr č. 6), druhým je pak takové poškození souboru, že jej nelze

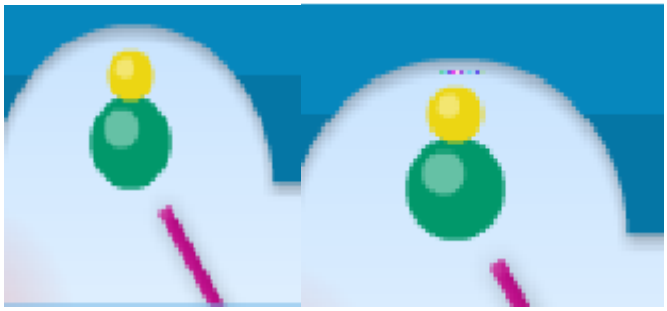
ani otevřít. To sice nebrání úspěšné extrakci skrytých dat, ale asi to ztrácí smysl z pohledu steganografie.

Obr č. 6: Výsledek ukrytí krátkého textového souboru metodou 2b3.

Obr č. 7: Výsledek ukrytí jiného krátkého textového souboru metodou 2b3.

Obr č. 8: Výsledek ukrytí krátkého textového souboru metodou 1b8.

Použití LSB i upravené verze b1 v BMP vyšlo podle očekávání. Na obrázcích nebyly patrné změny a upravená 1b vykazovala až o 12 % méně změněných bitů. Vkládání metodou 2b3 vedlo k viditelným lokálním změnám (Obr č. 9).



Obr č. 9: Výsledek ukrytí krátkého textového souboru metodou 2b3 (originál vlevo).

Použití na textové soubory dopadlo podle očekávání. V místě uložení dochází k nepřehlédnutelnému poškození textu.

5. DISKUSE

Postup zvolený v této práci pro zhotovení programu je spíše úvahou nad tím, zda je vůbec možné ubírat se takovou cestou. Problémy s časovou náročností během analýzy možností využití porovnávání řetězců dokazují, že je tento přístup hodně vzdálený od možnosti širšího využití. Také se zde naráží na všeobecný problém nutnosti přenášet se stego-médiem doplňující data. Je sice možné vytvořit steganografický program, pro který by nebylo nutné nic víc, než mít k dispozici stego-médium, ovšem jistě by to bylo za cenu toho, že jakmile by na tento program upřela pozornost steganalýza, pozbyl by svého smyslu.

Dalším nedostatkem je jistě počet vzorků, ze kterého vzešly výsledky analýzy. Na možnost učinit z výsledků nějaký obecnější závěr je jich velmi málo. Ovšem i tady narážíme na časovou náročnost prováděných porovnání.

K jistě lepšímu nastavení experimentu by vedla i hlubší znalost teorie pravděpodobnosti.

Výsledkem snažení je tak program, který ovládá klasickou metodu LSB a její upravenou verzi s nastavitelným krokem mezi bity. Za úspěch lze tedy povazovat případ, kdy je krok nastaven na 8. Za těchto podmínek se tedy provádí klasická metoda LSB s obohacením o vyhledání vhodného řetězce.

6. ZÁVĚR

Cílem této práce bylo seznámení se s problematikou digitální steganografie. Toho bylo dosaženo v teoretické části práce, kde byly popsány základní principy fungování, metody a využití, s přihlédnutím k souvislostem s některými dalšími rozsáhlými problematikami (digitální vodoznak, kryptografie). Součástí je také jedna kapitola doplňující výčet studií odkazovaných v textu a zmínka o dostupném software.

V praktické části je pak provedena analýza možností dosažení jednoho z hlavních cílů práce, vytvoření aplikace, která dokáže ukrývat a vyjímat zprávu (soubor) z krycího média, následná realizace vybrané myšlenky v podobě provedení experimentu a samotný postup vytvoření cílového programu. Ten byl pak podroben testu, ve kterém nejlépe obstála metoda LSB a ještě vylepšená verze s vyhledáváním vhodného řetězce 1b8 (viz kap. 4.2)

Vzhledem k nenalezení vhodného programu, se kterým by bylo možné porovnat výsledky z vytvořené aplikace, nebyla prováděna srovnávací analýza.

7. SEZNAM POUŽITÉ LITERATURY

[1] **MADOŠ, B.:** *Tajomstvá steganografie*, archív PC Revue, 3/2003.
Dostupný také z WWW:
http://www.gljs.sk/~sjiricek/inf/pcrevue/steganografia.pdf
[cit. 2008-1-25]

[2] **MORKEL, T.; ELOFF, J. H. P.; OLIVIER, M. S.:** *An Overview of Image Steganography*, Proceedings of the Fifth Annual Information Security South Africa Conference (ISSA2005), Sandton, South Africa, June/July 2005.
Dostupný také z WWW: http://mo.co.za/open/stegoverview.pdf
[cit. 2008-1-25]

[3] **MANGARAE, A.:** *Steganography FAQ*, [Zone-H.Org], 2006.
Dostupný také z WWW:
http://www.infosecwriters.com/text_resources/pdf/Steganography_AMangarae.pdf
[cit. 2008-1-25]

[4] **PROVOS, N.; HONEYMAN, P.:** *Detecting Steganographic Content on the Internet*, ISOC NDSS'02, San Diego, CA, February 2002. [August 2001, CITI Techreport]
Dostupný také z WWW:
http://www.citi.umich.edu/techreports/reports/citi-tr-01-11.pdf
[cit. 2008-1-25]

[5] **THAMPI, SABU M.:** *Information Hiding Techniques: A Tutorial Review*, ISTE-STTP on Betwork Security & Cryptography, LBSCE 2004.
Dostupný také z WWW:
http://arxiv.org/ftp/arxiv/papers/0802/0802.3746.pdf
[cit. 2009-3-10]

[6] **NAGESH H.R.;** Adarsh Rao Kordcal; Chandra Sekaran, K.: *DIGITAL STEGANOGRAPHY: SEEING THE UNSEEN*, Manipal Institute of Technology, Manipal, India, datum nenalezeno.
Dostupný také z WWW:
http://studentprogress.com:8080/uploads/cogrec/impnts/stegno-camera.pdf.38325.62179
[cit. 2008-1-25]

[7] **2ND LT. JAMES CALDWELL (U.S. AIR FORCE):** *Steganography*, CrossTalk The Journal of Defence Software engineering, Jun 2003 Issue.
Dostupný také z WWW:
http://www.stsc.hill.af.mil/Crosstalk/2003/06/caldwell.pdf
[cit. 2008-1-25]

[8] **WATKINS, J.:** *Steganography Messages Hidden in Bits*, 2001.
Dostupný také z WWW:
http://mms.ecs.soton.ac.uk/mms2002/papers/6.pdf
[cit. 2008-1-25]

[9] **CACHIN, CH.:** *Digital steganography*, Encyclopedia of Cryptography and Security. Springer, 2005.
Dostupný také z WWW:
http://www.zurich.ibm.com/~cca/papers/encyc.pdf
[cit. 2008-1-25]

[10] **JOHNSON, N. F.; JAJODIA, S.:** *Exploring Steganography: Seeing the Unseen*, Computer, vol. 31, no. 2, pp. 26-34, Feb., 1998.
Dostupný také z WWW: http://www.jjtc.com/pub/r2026.pdf
[cit. 2008-1-25]

[11] **ZHANG, W.; LI, SHIQU:** *Steganographic Codes -- a New Problem of Coding Theory*, Computer Science - Cryptography and Security, D.2.11, E.4, 2005.
Dostupný také z WWW:
http://arxiv.org/PS_cache/cs/pdf/0505/0505072v1.pdf
[cit. 2009-3-18]

[12] **LIN, E. T.; DELP, E. J.:** *A review of data hiding in digital images*, Proceedings of the conference on Image processing, image quality, image capture systems PICS '99, 25-28 April 1999, Savannah, Georgia, pp. 274-278.
Dostupný také z WWW:
ftp://skynet.ecn.purdue.edu/pub/dist/delp/pics99-stego/paper.pdf
[cit. 2008-1-25]

[13] **CUMMINS, J.; DISKIN, P.; ET AL.:** *Steganography And Digital Watermarking*, School of Computer Science, The University of Birmingham, 2004.
Dostupný také z WWW:
http://www.cs.bham.ac.uk/~mdr/teaching/modules03/security/students/SS5/Steganography.pdf
[cit. 2008-1-25]

[14] **MATTHEWS, CH.:** Behind The Music: *Principles Of Audio Steganography*, SE 4C03 Winter 2003.
Dostupný také z WWW:
http://www.cas.mcmaster.ca/~wmfarmer/SE-4C03-03/projects/papers/Matthews.pdf
[cit. 2008-1-25]

[15] **MAZURCZYK, W.; SZCZYPIORSKI, K.:** *Steganography of VoIP Streams*, On the Move to Meaningful Internet Systems: OTM 2008, 2008.
Dostupný také z WWW:
http://arxiv.org/ftp/arxiv/papers/0805/0805.2938.pdf
[cit. 2009-3-28]

[16] **WANG, YING; MOULIN, PIERRE:** *Perfectly Secure Steganography: Capacity, Error Exponents, and Code Constructions*, Information Theory, IEEE Transactions on, Volume: 54, June 2008.
Dostupný také z WWW:
http://arxiv.org/PS_cache/cs/pdf/0702/0702161v3.pdf
[cit. 2009-3-28]

[17] **DICKMAN, S. D.:** *An Overview of Steganography*, Report, James Madison University 2007, JMU-INFOSEC-TR-2007-002.
Dostupný také z WWW: http://www.infosec.jmu.edu/reports/jmu-infosec-tr-2007-002.pdf
[cit. 2008-1-25]

[18] **LENTI, J.:** *Steganographic method*, Per. Pol. Elec. Eng., 44/3-4 (2000), 249-258.
Dostupný také z WWW:
http://www.pp.bme.hu/ee/2000_3/pdf/ee2000_3_04.pdf
[cit. 2008-1-25]

[19] **ANDERSON, R. J.; PETITCOLAS, F. A. P.:** *On The Limits of Steganography*, IEEE Journal of Selected Areas in Communications, 16(4):474-481, May 1998, ISSN 0733-8716.
Dostupný také z WWW:
http://www.cl.cam.ac.uk/~rja14/Papers/jsac98-limsteg.pdf
[cit. 2008-1-25]

[20] **PROVOS, N.; HONEYMAN, P.:** *Hide and Seek: An Introduction to Steganography*, IEEE Security and Privacy, vol. 01, no. 3, pp. 32-44, May-June, 2003.
Dostupný také z WWW:
http://www.citi.umich.edu/u/provos/papers/practical.pdf
[cit. 2008-1-25]

[21] **KHARRAZI, M.; SENCAR, H. T.; MEMON, N.:** *Image steganography: Concepts and practice*, Lecture Notes Series, Institute for Mathematical Sciences, National University of Singapore, 2004.
Dostupný také z WWW:
http://isis.poly.edu/~steganography/pubs/ims04.pdf
[cit. 2008-1-25]

[22] **FRIDRICH, J.; GOLJAN, M.:** *Practical steganalysis of digital images - state of the art*, Proceedings of the SPIE Photonics West (Security and Watermarking of Multimedia Contents IV), Volume 4675, San Jose, California, USA (2002) 1—13.
Dostupný také z WWW:
http://www.ws.binghamton.edu/fridrich/Research/steganalysis01.pdf
[cit. 2008-1-25]

[23] **FARID, H.:** *Detecting steganographic message in digital images*. Technical Report TR2001-412, Dartmouth College, 2001.
Dostupný také z WWW:
http://www.ists.dartmouth.edu/library/dsm0401.pdf
[cit. 2008-1-25]

[24] **GOLJAN, M.; HOGEA, D.:** *Steganalysis of JPEG Images: Breaking the F5 Algorithm*, 5th Information Hiding Workshop, Noordwijkerhout, The Netherlands, 7–9 October 2002, pp. 310-323.
Dostupný také z WWW:
http://www.ws.binghamton.edu/fridrich/Research/f5.pdf
[cit. 2008-1-25]

[25] **AVCIBAS, I.; KHARRAZI, M.; ET AL.:** *Image Steganalysis with Binary Similarity Measures*, EURASIP JOURNAL ON APPLIED SIGNAL PROCESSING, 2005, NUMB 17, pages 2749-2757.
Dostupný také z WWW:
http://isis.poly.edu/~steganography/pubs/jasp_bsm.pdf
[cit. 2008-1-25]

[26] **SMETANA, M.; PENKALA, P.:** *Steganografie*, Kriminalistika, Roč. 39, č. 4 (2006), s. 246-251.
Dostupný také z WWW:
http://www.mvcr.cz/casopisy/kriminalistika/2006/04/steganografie.pdf
[cit. 2008-1-25]

[27] **GOLJAN, M.:** *Digital Image Steganography Using Stochastic Modulation*, in Proc. SPIE Electronic Imaging Santa Clara, CA, Jan 2003, pp. 191-202.
Dostupný také z WWW:
http://www.ws.binghamton.edu/fridrich/Research/stochastic_modulation02.pdf [cit. 2008-1-25]

[28] **HANSEN, K.; HAMMER, CH.; ET AL.:** *Steganographic Capacity of Images, based on Image Equivalence Classes*, ACM Multimedia '01, Workshop on Multimedia and Security, Ottawa Oct. 5, 2001, ISBN 1-58113-393-6).
Dostupný také z WWW: http://www.randleff.dk/stego/hansen-infocap.pdf
[cit. 2008-1-25]

[29] **KHARRAZI, M.; SENCAR, H. T.; MEMON, N.:** *Benchmarking steganographic and steganalysis techniques*, Security, Steganography, and Watermarking of Multimedia Contents VII. Proceedings of the SPIE, Volume 5681, pp. 252-263 (2005).
Dostupný také z WWW:
http://isis.poly.edu/~steganography/pubs/spie05.pdf
[cit. 2008-1-25]

[30] **BERG, G.; DAVIDSON, I.; ET AL.:** *Searching for Hidden Messages: Automatic Detection of Steganography*, Proceedings of the 2007 international workshop on Domain driven data mining, California, p. 24 – 32, 2007, ISBN:978-1-59593-846-6.
Dostupný také z WWW:
http://www.cs.albany.edu/~davidson/Publications/IAAI103.pdf
[cit. 2008-1-25]

[31] **HO, A.; TAM, S-C.; ET AL.:** *Digital Steganography for Information Security*, 1999.
Dostupný také z WWW: http://www.datamark-tech.com/pdf/steganography.pdf
[cit. 2008-1-25]
[cit. 2008-1-25]

[32] **RAKAN EL-K.; ANGELOS D. K.:** *Hydan: Hiding Information in Program Binaries*. ICICS 2004: 187-199.
Dostupný také z WWW:
http://www1.cs.columbia.edu/~angelos/Papers/hydan.pdf
[cit. 2008-1-25]

[33] **FRIDRICH, J.; GOLJAN, M.; SOUKAL, D.:** *Higher-order statistical steganalysis of palette images*, Proceedings – SPIE the international sociaty for optical engineering, 2003, ISSU 5020, pages 178-190.
Dostupný také z WWW:
http://www.ws.binghamton.edu/fridrich/Research/pairs01.pdf
[cit. 2008-1-25]

[34] **CHAE, J. J.; MANJUNATH, B. S.:** *Data Hiding in Video*, The Pennsylvania State University CiteSeer Archives, 1999.
Dostupný také z WWW:
http://vision.ece.ucsb.edu/publications/99ICIP.pdf
[cit. 2008-1-25]

[35] **RABAH, K.:** *Steganography-The Art of Hiding Data*, Information Technology Journal 3 (3): 245-269, 2004, ISSN 1682-6027.
Dostupný také z WWW::
http://docs.ksu.edu.sa/PDF/Articles17/Article170588.pdf
[cit. 2008-1-25]

[36] **FRIDRICH, J.; GOLJAN, M.; HOGEA, D.:** *New methodology for breaking steganographic techniques for JPEGs*, Proc. SPIE, vol.5020, pp.143–155, 2003.
Dostupný také z WWW:
http://www.ws.binghamton.edu/fridrich/Research/jpeg01.pdf
[cit. 2008-1-25]

[37] **HOPPER, N. J.; LANGFORD, J.; VON AHN, L.:** *Provably Secure Steganography*, Proceedings of the 22nd Annual International Cryptology

Conference on Advances in Cryptology, p. 77 - 92, 2002, ISBN:3-540-44050-X.
Dostupný také z WWW: http://www.cs.cmu.edu/~biglou/PSS.pdf
[cit. 2008-1-25]

[38] **MICHE, Y.; ROUE, B.; ET. AL.:** *A Feature Selection Methodology for Steganalysis*, in Lecture Notes in MRCS – Multimedia Content Representation, Classification and Security, International Workshop, 2006: 49-56, Istambul .
Dostupný také z WWW:
http://www.cis.hut.fi/projects/tsp/Publications/Publication69.pdf
[cit. 2008-1-25]

[39] **ANDERSON, R. J.:** *Stretching the Limits of Steganography*, in Information Hiding, Springer Lecture Notes in Computer Science, 1996, vol. 1174, pp. 39--48.
Dostupný také z WWW:
http://www.cl.cam.ac.uk/~rja14/Papers/stegan.pdf
[cit. 2008-1-25]

[40] **MOSKOWITZ, IRA S.; LONGDON, GARTH E.; CHANG, LIWU:** *A New Paradigm Hidden in Steganography*, in New Security Paradigms Workshop 2000, publikováno 2001, Irsko, ACM Press, pp. 41-50, ISBN:1-58113-260-3.
Dostupný také z WWW:
http://chacs.nrl.navy.mil/publications/CHACS/2000/2000moskowitz-stego.pdf
[cit. 2008-1-25]

[41] MAZURCZYK, W.; KOTULSKI, Z.: New *security and control protocol for VoIP based on steganography and digital watermarking*, Annales UMCS, Informatica, AI 4 (2006), ISNN 1732-1360.
Dostupný také z WWW:
http://www.ippt.gov.pl/~zkotulsk/IBIZA_2006_en.pdf
[cit. 2008-1-25]

[42] **HAIRONG QI; SNYDER, W.E.; SANDER, W.A.:** *Blind consistency-based steganography for information hiding in digital media*, Multimedia and Expo, ICME '02. 585- 588 vol.1, Digital Object Identifier 10.1109/ICME.2002.1035849.
Dostupný také z WWW:
http://aicip.ece.utk.edu/publication/02icme.pdf
[cit. 2008-1-25]

[43] **RU XUE-MIN; ZHUANG YUE-TING; WU FEI:** *Audio steganalysis based on "negative resonance phenomenon" caused by steganographic tools*, Journal of Zhejiang University - Science A, ISSN 1862-1775, 577-583, 2006.
Dostupný také z WWW:
http://engine.cqvip.com/content/tp/88140x/2006/007/004/gc57_tp2_21526145.pdf
[cit. 2008-1-25]

[44] **AHSAN, K.; KUNDUR, D.:** *Practical data hiding in TCP/IP*, in Proc. ACM Workshop on Multimedia Security, 2002.
Dostupný také z WWW:
http://www.ece.tamu.edu/~deepa/pdf/acm02.pdf
[cit. 2008-1-25]

[45] **TRIVEDI, S.; CHANDRAMOULI, R.:** *Active steganalysis of sequential steganography*. SPIE conference California, 5020(13):123--130, January 2003.
Dostupný také z WWW: http://www.ece.stevens-tech.edu/~mouli/activesteg.pdf
[cit. 2008-1-25]

[46] **MAZURCZYK, W.; KOTULSKI, Z.:** *Covert channel for improving VoIP security, Advances in Information Processing and Protection*, pp.271-280, Springer, Berlin 2007. ISBN: 978-0-387-73136-0 (Print) 978-0-387-73137-7 (Online).
Dostupný také z WWW:
http://www.ippt.gov.pl/~zkotulsk/Covert%20Channel%20for%20Improving%20VoIP%20Security.pdf
[cit. 2008-1-25]

[47] **PROVOS, N.:** *Defending against statistical steganalysis*, Proceedings of the 10th conference on USENIX Security Symposium - Volume 10, 24 - 24 , Washington D.C. 2001.
Dostupný také z WWW:
http://www.citi.umich.edu/u/provos/papers/defending.ps
[cit. 2008-1-25]

[48] **TZSCHOPPE, R.; BAEUML, R.; ET AL.:** *Steganographic system based on higher-order statistics*, Proceedings- SPIE The International Society for Optical Engineering, 2003, ISSU 5020, pages 156-166.
Dostupný také z WWW:
http://www.lnt.de/LIT/papers/ei2003_stego.pdf
[cit. 2008-1-25]

[49] **HARMSEN, J. J.; PEARLMAN, W. A.:** *Steganalysis of additive noise modelable information hiding*, in Proc. SPIE Electronic Imaging 5022, (Santa Clara, CA), Jan. 21--24, 2003.
Dostupný také z WWW:
http://www.cipr.rpi.edu/~pearlman/papers/harmsEI03.pdf
[cit. 2008-1-25]

[50] **GOU, H.M.[HONG-MEI];** Swaminathan, A.[Ashwin]; et al.: *Noise Features for Image Tampering Detection and Steganalysis*, ICIP07(VI: 97-100), Image Processing 2007, ISBN: 978-1-4244-1437-6, ISSN: 1522-4880.
Dostupný také z WWW:
http://www.ece.umd.edu/~ashwins/pdf/ICIP07a.pdf
[cit. 2008-1-25]

[51] **MCDONALD, A.; KUHN, M.:** *StegFS: A Steganographic File System for Linux*, Lecture Notes In Computer Science; Vol. 1768 , Proceedings of the Third International Workshop on Information Hiding , Pages: 462 - 477 , 1999 , ISBN:3-540-67182-X.
Dostupný také z WWW: http://www.cl.cam.ac.uk/~mgk25/ih99-stegfs.pdf
[cit. 2008-1-25]

[52] ZHONG, S.; CHENG, X.; CHEN, T.: *Data Hiding in a Kind of PDF Texts for Secret Communication*, International Journal of Network Security, Vol.4, No.1, PP.17–26, Jan. 2007.
Dostupný také z WWW: http://ijns.nchu.edu.tw/contents/ijns-v4-n1/ijns-2007-v4-n1-p17-26.pdf
[cit. 2008-1-25]

[53] **GILANI,S. A. M.; SKODRAS, A. N.:** *DLT-Based Digital Image Watermarking*, Proc. First IEEE Balkan Conference on Signal Processing, Communications, Circuits and Systems, Istanbul, Turkey, June 1-3, 2000.
Dostupný také z WWW:
http://www.ee.bilkent.edu.tr/~signal/BCSP/skodras.pdf
[cit. 2008-1-25]

[54] **STANEV, S.:** *Steganographic Systems*, CSC/MAT 494.
Dostupný také z WWW:
http://www.nku.edu/~mcsc/mat494/uploads/StanevPaper.pdf
[cit. 2008-1-25]

[55] **ZHOU, X.; PANG, H-H.; TAN, K-L.:** *Hiding Data Accesses in Steganographic File System*, Proceedings of the 20th International Conference on Data Engineering, p.572, March 30-April 02, 2004 .
Dostupný také z WWW: http://dataquality.i2r.a-star.edu.sg/hhpang/publications/StegFS-traffic.pdf
[cit. 2008-1-25]

[56] **TOSO, M.; CHUNG, S. B.:** *Combining Steganography and Zero-Knowledge Proofs to Embed and Prove a Digital Signature in an Image*, Quantitative Methods and Computer Science, University of St. Thomas, Saint Paul, MN 55105.
Dostupný také z WWW:
http://www.micsymposium.org/mics_2006/papers/TosoAndChung.pdf
[cit. 2008-1-25]

[57] **HOSMER, CH.:** *Discovering Hidden Evidence*, Journal of Digital Forensic Practice, Volume 1, Issue 1, pages 47 – 56, 2006.
Dostupný také z WWW:
http://www.wetstonetech.com/f/Stego_Article_0707.pdf
[cit. 2008-1-25]

[58] **MUNRO, K.:** *Steganography - is it becoming a double-edged sword in computer security?*, University of the Witwatersrand, Conference ISSA 2002 (Information Security for South Africa).
Dostupný také z WWW:
http://icsa.cs.up.ac.za/issa/2002/proceedings/A013.pdf
[cit. 2008-1-25]

8. KLÍČOVÁ SLOVA

digitální steganografie
ukrývání dat
krycí médium
stgo-médium
LSB
steganalýza
digitální vodoznak

digital steganography
data hiding
cover medium
stego-medium
LSB
steganalysis
digital watermarkings

9. PŘÍLOHY

Příloha A: Program Test GC pro porovnávání řetězců – přiložené CD

Příloha B: Program GetConcealed – přiložené CD

Příloha C: Stego-média vytvořená programem GetConcealed – přiložené CD